MEKON

FrameAC

User Guide

Version 4.3

March 2007

SOFTWARE END USER LICENCE AGREEMENT

Before you open the package containing the software media, you should carefully read the Terms and Conditions of this Licence agreement. By opening the package, you are consenting to be bound by and are becoming a party to this licence agreement. If you find any of the terms objectionable, please contact Mekon Limited, or return the package UNOPENED within 30 days of purchase for a refund.

This licence is not an agreement for sale. Mekon Limited continues to own the copy of the software and the physical media contained in this package and any other copy that you are authorised to make pursuant to this agreement.

1.Use of the Software

You may install the software in a single geographical location such that the number of users with local or temporary remote access to the software does not exceed the permitted number of users purchased and indicated on the associated sales invoice.

The software can be installed on Windows, however, the warranty will only apply to those operating systems described as compatible in the user documentation.

Should access by additional users be required, please contact Mekon for written permission.

Home Use - The primary user of each computer on which the software is installed or used, may also install the software on one home or portable computer. However, except for the purposes of demonstration, the portable computer may not be used on another business site.

2.Copyright

The software is the Intellectual Property of Mekon Limited and is protected by copyright law. Should the software or any part of it be reproduced, utilised in a modified form or incorporated into another software package without prior written agreement, legal action will be taken against any user, reseller, individual or company producing such software.

A single copy of the media may be made and used in case of back-up and recovery.

All documentation is also subject to copyright law and may not be copied or reproduced in any form.

3.Modification

The software must not be modified in any way without the express permission of Mekon Limited.

4.Source Code

You agree not to reverse engineer, decompile, disassemble or otherwise attempt to discover the source code of the software except as expressly permitted by Mekon Limited. The source code for the software is not normally provided, but may be purchased subject to negotiation by contacting Mekon Limited.

5.Transfer

You may not rent, lease, sublicense or lend all or part of the software or documentation. You may however, request a to transfer all your rights to use the software and documentation to another person or legal entity provided that (i) you transfer this agreement, the software, including all copies, updates and prior versions and documentation to such person or entity. (ii) your retain no copies, including copies stored on a computer, (iii) you retain no modified copies of the software if an EDD is included in the software, (iv) that the receiving party agrees to be bound by the terms and conditions of the agreement, (v) that the software is covered by a valid Mekon Limited maintenance contract at the time of transfer.

6.Limited Warranty

Mekon warrant that the software will perform substantially in accordance with the documentation for the ninety (90) day period following your receipt of the software. No liability for any damages arising out of the use of the software is accepted by Mekon.

This Licence Agreement is governed by the laws of England.

If you have any questions regarding this agreement, please contact Mekon Limited.

USER GUIDE END USER LICENCE AGREEMENT

This guide, as well as the software described in it, is furnished under license and may only be used or copied in accordance with the terms of such license. The information in this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Mekon Limited. Mekon Limited assumes no responsibility or liability for any errors or inaccuracies that may appear in this book.

Except as permitted by such license, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Mekon.

Please remember that existing artwork or images that you may desire to scan as a template for your new image may be protected under copyright law. The unauthorised incorporation of such artwork or images into your new work could be a violation of the rights of the author. Please be sure to obtain any permission required from such authors.

This Licence Agreement is governed by the laws of England.

If you have any questions regarding this agreement, please contact Mekon Limited.

TRADEMARKS AND COPYRIGHTS USED IN SOFTWARE AND DOCUMENTATION

Adobe, Frame, FrameMaker are registered trademarks of Adobe Systems, Inc.

Windows, Excel, Word and Access are registered trademarks of Microsoft Corporation.

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

Written and designed at

Mekon Ltd

31–35 St. Nicholas Way

Sutton

Surrey

SM1 1JN

United Kingdom

Table of Contents

Chapter 1 *Getting Started with FrameAC*

Introduction 1

Initial Concepts 2

- Types of FrameAC Applications 2
- FrameAC as a Bridge to FrameMaker 2
- Objects, Events, Methods, and Properties 3

Installing FrameAC 4

- Evaluation License Restrictions 4
- Temporarily Disabling FrameAC 4
- Removing FrameAC 5

Technical Support 5

Chapter 2 *FrameAC Tutorial*

Configuring FrameAC 7

- FrameAC Licensing Options 7
- Configuring the maker.ini FIle 7

Tutorial: Creating a FrameAC Plug-in for Visual Basic .Net 8

- Creating and Configuring a New .Net Project 8
- Browsing the FrameAC libraries 10
- Registering the Plug-In 14
- Debugging the Plug-in 14
- Final Installation 15

Tutorial: Creating a FrameAC Controller App in Visual Basic .Net 15

- Create a New Project 15
- Writing the Code 17
- Registering the Application 19
- Debugging the Application 19
- Final Installation 19

Tutorial: Creating a FrameAC Plug-in for Visual Basic 6 20

- Creating and Configuring a New Project 20
- Browsing the FrameAC libraries 21
- Writing the Code 22
- Registering the Plug-in 25
- Debugging the Plug-in 25
- Final Installation 26

Tutorial: Creating a FrameAC Controller App. 26

- Creating a New Project 27
- Writing the Code 27
- Registering the Application 29
- Debugging the Plug-in 29
- Final Installation 30

Tutorial: Creating FrameAC Scripts 30

- Visual Basic Script 31
- JavaScript 32
- Event-triggered Script 32

Tutorial: Calling script from Visual Basic 33

- Preparing the Visual Basic module 33
- Executing the script 34
- Executing the script with parameters 34

Chapter 3 *Object Reference*

FMAnchoredFrame 37
FMArc 40
FMBodyPage 40
FMBook 42
FMBookComponent 58
FMCell 67
FMCharacterFormat 70
FMColor 75
FMCombinedFontDefinition 77
FMCommand 78
FMCommandObject 84
FMConditionalFormat 85
FMCrossReference 86
FMCrossReferenceFormat 87
FMDocument 88
FMElement 129
FMElementDefinition 138
FMEllipse 141
FMFlow 141
FMFootnote 142
FMFormatChangeList 143
FMFormatRule 152
FMFormatRuleClause 153
FMGraphic 155
FMGroup 159
FMHiddenPage 159
FMInset 160
FMLine 161
FMMarker 162
FMMarkerType 162
FMMasterPage 163
FMMath 164
FMMenu 165
FMMenuItemSeperator 166
FMObject 166
FMPage 168
FMPageFrame 168
FMParagraph 169
FMParagraphFormat 179
FMPolyLine 189
FMPolygon 190
FMRectangle 190
FMReferencePage 191
FMRoundRectangle 191
FMRow 191
FMRubi 193
FMRulingFormat 194
FMSession 195
FMSubCol 203
FMTable 204
FMTableFormat 210
FMTextFrame 214
FMTextInset 216
FMTextInset_ApiClient 217
FMTextInset_Flow 218
FMTextInset_Text 219
FMTextInset_TextTable 220
FMTextLine 221
FMUnanchoredFrame 224
FMVariable 224
FMVariableFormat 225

Chapter 4 *Event Reference*

OnApiCommand 227
OnApiMessage 229
OnApiNotify 230

Chapter 5 *Method Reference*

AddCols 237
AddCommandToMenu 238
AddMenuToMenu 239
AddRows 241
AddText 242
Alert 243

ApplyPageLayout 244
BailOut 245
CallClient 246
CenterOnText 248
Clear 248
ClearAllChangebars 250
Close 251
Compare 252
Copy 253
CustomDoc 255
Cut 256
DefineCommand 257
DefineMenu 259
Delete 260
DeleteBodyRows 261
DeleteCellText 261
DeleteCols 262
DeleteRowText 263
DeleteRows 264
DeleteText 265
DeleteTextInsetContents 266
DeleteUndefinedAttribute 267
DemoteElement 267
ElementLocToTextLoc 268
ExecuteScript 269
ExecuteScriptWithParam 269
Find 270
Generate 274
GetCountOfRowsOfType 275
GetElementCatalog 276
GetFirstRow 277
GetImportDefaultParams 277
GetLastRow 283
GetNamedObject 284
GetObject 285
GetOpenDefaultParams 286
GetPropVals 292
GetSaveDefaultParams 293
GetText 298
GetTextForRange 304
GetTextProps 306
GetUpdateDefaultParams 307
Import 309
ImportElementDefs 313
ImportFormats 314
Initialise 315
IsCellEmpty 316
IsRowEmpty 317
MakeTblSelection 318
MenuItemInMenu 319
MergeIntoFirst 320
MergeIntoLast 321
NewAnchoredFormattedObject 321
NewAnchoredObject 322
NewBodyPage 323
NewBook 324
NewBookComponent 325
NewBookComponentInHierarchy 326
NewChild 328
NewDocument 329
NewElement 330
NewElementInHierarchy 331
NewFormatChangeList 333
NewFormatRule 333
NewFormatRuleClause 334
NewNamedObject 335
NewParagraph 337
NewTable 338
ObjectValid 340
OpenBook 340
OpenDocument 346
Paste 352
PopClipboard 354
Print 355
PromoteElement 356
PushClipboard 356
Redisplay 357
Reformat 358
Rehyphenate 358
ResetEqnSettings 359
ResetReferenceFrames 359
RestartPgfNumbering 360
Save 361
ScrollToText 362
SetNotification 363
SetPropVals 370
SetTextProps 372
SimpleOpenBook 373
SimpleOpenDocument 374
SimpleSave 375
SplitElement 376
StraddleCells 376
TextLocToElementLoc 377
UnStraddleCells 378
UnWrapElement 380
UpdateBook 381

UpdateTextInset 383
UpdateVariables 384
UpdateXRefs 384
WrapElement 385

Chapter 6 *Data Type Reference*

F_Int 387
F_Metric 387
F_String 389
F_TypedValValues 389
udtAttribute 390
udtAttributeDef 391
udtCompareRet 392
udtElementCatalogEntry 392
udtElementLoc 393
udtElementRange 394
udtFont 394
udtPoint 395
udtPropIdent 395
udtPropVal 396
udtTab 396
udtTextItem 397
udtTextLoc 400
udtTextRange 401
udtTypedVal 401

Appendix A *Special Cases for Object Properties 403*

FMDocument Hypertext Properties 403
HypertextParseError 403
HypertextValidateError 404
Command Codes 405
Link Destinations 406
Link Destination Object Types 406
FMGraphic properties 407
Pen and FIll properties 407
Dash Patterns 407
Graphic and Text Inset Hint Strings 407
Graphic Inset Import Hint Strings 407
Text Inset Import Hint Strings 410

Appendix B *FrameMaker Document and Session Architecture 413*

How FrameMaker Represents Text 413
How FrameAC Represents Font Information 417
Standard Marker Types for a Document 419
XMP Data in FrameMaker Documents 419
PDF Document Info Dictionaries 421

1 Getting Started with FrameAC

Welcome to FrameAC, the Visual Basic development and connectivity tool for FrameMaker. FrameAC provides the means for developers to enhance the functionality of FrameMaker via programs in Visual Basic (VB), and scripts in Visual Basic Script and JavaScript.

Whether you are a seasoned FDK programmer or new to developing FrameMaker enhancements, we think you will find many advantages to the FrameAC environment. It is our hope that you enjoy using FrameAC, and find it a productive and pleasurable environment to work in.

Introduction

FrameAC is a development tool similar to the Frame Development Kit (FDK), which is an API developed for the C programming language. The main difference between FrameAC and the FDK is that FrameAC programs are written in Visual Basic. FrameAC provides nearly all the coverage that is provided by the FDK, — whatever coverage FrameAC does not provide is taken care of by virtue of the Visual Basic environment(6 or .Net). Examples include the following:

- Dialog box management — The FDK includes routines for displaying and responding to dialog boxes. VB includes dialog box creation, management, and event response. In addition, VB dialog boxes use the full range of Windows controls, and can include custom controls.
- Memory management — VB handles all allocation and deallocation, so there is no need for memory management routines.
- File management and OS interaction — VB includes its own set of routines to operate on files and interact with the OS.
- List and string routines — The FDK includes lists of strings, integers and other data types. FrameAC uses arrays of Variant data to accomplish the same things. Also VB includes its own string routines.

There are other advantages to developing FrameMaker enhancements in Visual Basic. The language is in wide use, and standardised. The language syntax streamlines many operations such as getting and setting object properties, or establishing a location in a document. With VB you have easy access to COM and ActiveX technologies. You can create full VB applications for which connections to FrameMaker are an integral part. You can connect FrameMaker to other products that publish libraries you can reference in Visual Basic. And you can use VBA (Visual Basic for Applications) to call FrameMaker from within other applications that support that technology.

Initial Concepts

FrameAC is a rich and powerful product. You can use it to customize your FrameMaker workflow in many ways. Before you get started, you should understand what FrameAC is, and how it relates to FrameMaker.

Types of FrameAC Applications

You can use FrameAC to develop three types of applications:

- Runtime scripts
- ActiveX plug-ins
- Controller applications

You create runtime scripts within the FrameAC scripting environment. These scripts are best for handling repetitive tasks that involve a number of direct steps. You can develop them to run on a command or keystroke, or to run in response to specific user events such as opening or closing a file, clicking the mouse, or quitting the FrameMaker application.

ActiveX plug-ins are applications you create and then register in the maker.ini file. As FrameMaker starts up, it looks for registered ActiveX plug-ins to run, and starts them as FrameMaker initialises. You create ActiveX plug-ins in the Visual Basic environment. A typical use for plug-ins is to add functions to FrameMaker, and post menu commands to execute these functions. In that sense they're similar to scripts, but plug-ins are more suitable for procedures that are complex, or interact with the OS to manipulate files or communicate with other applications.

Controller applications are .exe files that stand alone and run in their own space. However, they establish communication with FrameMaker and control actions within the FrameMaker session. FrameMaker must be running before you start the controller application — the controller application connects with the current FrameMaker session.

You create controller applications in the Visual basic environment. They are a good way to create an application that uses FrameMaker for printed output. For example, a custom database might use FrameMaker for reports.

You can also communicate with FrameMaker from a VBA process. For all intents and purposes, such a connection qualifies as a controller application. From the point of view of development, the coding uses the same techniques to establish the connection and perform actions in FrameMaker.

FrameAC as a Bridge to FrameMaker

FrameAC is a dynamically linked library (DLL) that connects FrameMaker to programs written in VB. It uses the FrameMaker API to connect with FrameMaker, and it uses the Microsoft Windows COM architecture to connect with the applications you create. In this

sense, FrameAC is a FrameMaker plugin that acts as a bridge between FrameMaker and your applications.

FrameAC also includes VBS and JavaScript interpreters that run alongside the FrameMaker process. These interpreters can run scripts you specify via commands that FrameAC adds to the FrameMaker menus. You do not need a VB development environment to create these scripts — FrameAC provides the scripting environment for you.

Frame Developer Kit (FDK) developers will be interested to know that FrameAC is the plug-in that communicates with FrameMaker — all the FrameAC applications a developer registers to run with FrameMaker are marshalled by the FrameAC DLL. This has consequences. In the first place, it's necessary to register FrameAC in the `maker.ini` file.

Another consequence is that all events and notifications that a FrameAC application might request pass through the FrameAC DLL, and are then sent to the individual applications. This is made apparent when handling OnApiMessage events. These events respond to users clicking on hypertext links with the following syntax:

```
message [plug-in name] [message string]
```

For the message to reach any FrameAC application, the plugin-name must be `FrameAC`. When such a message link is clicked, the message is sent to all FrameAC applications. The most important point of this is to realize that FrameAC is a FrameMaker plug-in, and it communicates with your applications.

Objects, Events, Methods, and Properties

FrameAC supports Visual Basic 6, Visual Basic .Net and JavaScript — three object-oriented programming languages. A FrameMaker session is treated as an object, as are all the things that can be created in a session, from documents and books to graphics, paragraphs, and format definitions.

Each object includes a number of parameters that describe it. For example, a document object includes a name parameter that describes the full path to that document file. A paragraph format object includes properties to express its font, its tabs, spacing above and below, and many other aspects about the paragraph format. Many properties are read-only — FrameMaker sets these properties, and changing them via FrameAC could result in a corrupt document. Other properties you can change, and this is one way to affect changes in a document. For example, you can change the font used for a paragraph. Objects and their properties are documented in Chapter 3, "Object Reference."

Objects also include methods that can perform specific actions. Examples include adding a new object into a series of objects, adding new structure elements, inserting text into a document, getting text from an object or selection range, opening, closing, and saving files, and many other actions. Methods are documented in Chapter 5, "Method Reference."

FrameAC uses events to trigger actions in your programs. These events are for initialisation, hypertext messages to your programs, and general notificatons. These events are especially

important for plug-ins and event-triggered scripts. Events are documented in Chapter 4, “Event Reference.”

The tutorials in this chapter necessarily make use of objects, events, methods and properties. If you are familiar with Visual Basic, then these should look familiar to you.

Installing FrameAC

Once you have downloaded FrameAC, follow these steps to install it on your computer:

1. Navigate to the location where you saved the FrameAC installation file, `Redist.exe`.
2. Double click the icon for `Redist.exe`.

   The InstallShield© wizard appears. Follow the instructions on the screen.
3. When the Licence dialog box appears, enter your information.

In this field:	Enter:
User name	Your user name
Company name	The company name agreed upon in your license agreement. For an evaluation license, enter "Evaluation"
Runtime license key	The key provided to you by Mekon. For an evaluation license, enter 2-7534-2021-3424-0079-0406
Design license key	The key provided to you by Mekon. For an evaluation license, enter 2-315B-5825-0338-542E-0079-0406

4. Click Finish

   FrameAC is now installed on your computer.

> **Important:** If the licence dialog is not displayed during installation. Then navigate to the FrameAC installation directory and run(double click) Licence.exe

Evaluation License Restrictions

FrameAC automatically starts each time you open Adobe FrameMaker. The evaluation licence allows FrameAC to run unlimited for 30 days from first installation.

Temporarily Disabling FrameAC

You can determine that FrameAC will not run when you start up FrameMaker. To do this, you edit the `maker.ini` file as follows:

1. Open your `maker.ini` file in a text editor.

   You can find the `maker.ini` file in your FrameMaker installation directory (usually inside the Program Files folder).

2. In maker.ini, find the `[api clients]` section.
3. Within the [api clients] section of the `maker.ini` file, find the following line:

   ```
   FrameAC=Standard, Mekon FrameAC, C:\Program Files\Mekon \FrameAC\FRAMECOM.DLL, all
   ```

4. Comment out this line by placing a semi-colon in front of the first character in the line, then restart FrameMaker.

   This causes FrameMaker to skip over FrameAC the next time it starts up.

5. To use FrameAC again, you must go to this line in the `maker.ini` file and remove the semi-colon, then restart FrameMaker.

Removing FrameAC

To remove FrameAC completely from your computer:

1. Open the Control Panel on your system.
2. Start the Add/Remove Programs utility.
3. Select FrameAC and click Change/Remove

   The InstallShield© wizard appears.

4. Select Remove, then click Next.
5. Click OK, then click Finish to remove all the FrameAC components.

Technical Support

At Mekon, we provide a range of support options to suit our customer needs. FrameAC support is offered as a paid option in blocks of four hours. Technical consultants are available to provide assistance to users. There is a users forum at http://tech.groups.yahoo.com/group/frameac/, we also offer the option of a customised support contract that can be adjusted to suit your particular needs. For more information on our support contract options, please contact the Mekon Sales team at info@mekon.com or Tel +44 (0) 20 8722 8400.

2 FrameAC Tutorial

FrameAC is an environment that provides programming access to Visual Basic (VB6. DotNet), Visual Basic for Applications (VBA), Visual Basic Script (VBS) and JavaScript programs or scripts. This tutorial describes how to control FrameMaker sessions, documents, and books via programs written in these languages. When you finish the tutorial, you should know how to:

- Use VB to create a plug-in — an ActiveX executable that can operate with FrameMaker as a plug-in
- Use VB to create a controller application — an executable that launches separately from FrameMaker, but connects with a FrameMaker session and controls actions within that session
- Use the FrameAC scripting environment to create VBS and JavaScript code that responds to user events and performs actions on FrameMaker documents

Configuring FrameAC

Before you get started with the tutorials, FrameAC must be properly configured. If you correctly installed FrameAC, you should be able to go straight to developing custom FrameMaker solutions. However, you should review the following information about licensing and .ini configuraton.

FrameAC Licensing Options

FrameAC has two licences — one for users and one for developers. With the user's licence, you can run any FrameAC applications, and you can create scripts in the FrameAC scripting environment. To create FrameAC plug-ins or controller applications, you must have a developer's licence. However, the current version of FrameAC includes both licences within the purchase price.

Configuring the maker.ini FIle

Once you have installed FrameAC and established your licence, you must have the proper entries in your maker.ini file. While FrameMaker supports separate maker.ini files for each user profile, it is recommended that you use the main maker.ini file, which is found in the FrameMaker installation directory.

[APIClients]
The ApiClients section of the .ini file must have the following entry:

```
FrameAC=Standard, Mekon FrameAC, <InstallDir>\Mekon\FrameAC\FRAMECOM.DLL,all
```

where `<InstallDir>` is the location of the FrameAC installation.

[FrameCOMClients]
The .ini file must also include a `[FrameCOMClients]` section. This is where you will register your FrameAC controller applications and plug-ins.

To register all your controller applications, you add the following entry once:

```
FrameAC.FrameEx={E6E0DCF4-A895-11D3-91B3-0050BAAE27FB}
```

You can find explicit instructions in the tutorial section for controller applications.

To register plug-ins, you add the following entry for each one:

```
<ProjectName>.<ClassName>={xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}
```

where `<ProjectName>` is the name of the plug-in project, and `<ClassName>` is the name of the plug-in class. You can find explicit instructions in the tutorial section for creating FrameAC plug-ins.

Tutorial: Creating a FrameAC Plug-in for Visual Basic .Net

This tutorial leads you through the steps to create a basic FrameAC plugin in Visual Basic .Net. These steps include:

- Creating a new VB.Net project and designing the application's form
- Writing the code
- Registering the plug-in
- Debugging the plug-in
- Final Installation
- Considerations

You will create this plug-in in Visual Basic .Net. The code uses events, methods and properties that may not be familiar to you. For more information about specific events, methods and properties please refer to the appropriate chapters of this manual.

Creating and Configuring a New .Net Project

FrameAC plug-ins created using .Net are COM components, and need to be registered for COM Interoperability. However with VB.Net unlike VB6 there is no requirement for the project that contains forms to be an EXE, it can be a DLL.

To create and configure the new project:

1. Create new Project.

   Start Visual Studio .Net and create a Visual Basic Class Library project. Giving it a name of `NetBasicTutorial`

2. Add References to the FrameAC libraries.

   Choose Project > Add Reference to open the Reference dialog box.

   Select the COM tab

   Then find the following entries and click '*select*'

   - Mekon FrameAC Type Library
   - Mekon FrameAC Library
   - Mekon FDK Types
   - Mekon FrameAC UDT Type Library

   Then click OK.

3. Specify the name of the plug-in class

   In the Filename field of the properties widow of `class1` set the name to `clsNetBasicTutorial.vb`

   In the code for clsNetBasicTutorial.vb change the class1 to clsNetBasicTutorial

   ```
   Public Class Class1
   ```

   becomes

   ```
   Public Class clsNetBasicTutorial
   ```

4. Add COM Interop Services to the class

   We need to add interop services to the class that we created to allow it to work as a COM client.

   Add the following lines to top of the code for clsNetBasicTutorial.vb

   - Imports System.Runtime.InteropServices
   - <System.Runtime.InteropServices.ProgId("NetBasicTutorial.clsNetBasicTutorial")> _

   Your code for this class so far should look like this:

   ```
   Imports System.Runtime.InteropServices
   <System.Runtime.InteropServices.ProgId("NetBasicTutorial.clsNetBasicTutorial")>_
   Public Class clsNetBasicTutorial

   End Class
   ```

5. Set the Project for COM Interop

   Choose Project > NetBasicTutorial Properties to open the properties dialog.

   Choose Configuration Properties > Build on the dialog

   Ensure that the *Register for COM Interop* checkbox has been ticked.

   Click Ok

   Note! you need to do this for both Debug and Release modes of the IDE

6. Save the project

You should save the project with a name and location that makes sense to you.

Important: Before you will be able to debug the FrameAC Plug-In that you have created you will have to copy the interop files and pluginui.dll files.

Before debugging the Plug-in for the first time perform the following steps.

1. Build the FrameAC Plug-in

   Choose Build > Build Solution

2. Copy the interop files

   Copy all the files prefixed with *Interop* located in the *project_directory\bin* directory to the *FrameMaker_Installation_Directory*.

3. Copy pluginui.dll

   Copy pluginui.dll from *FrameAC_Installation_Directory* to the *FrameMaker_Installation_Directory*.

The project is now configured to have access to all the FrameAC events, methods, objects and data types. You have also indicated that the project needs to be registered for COM interop and have provided the name that you will use to register the plug-in in the .ini file.

Browsing the FrameAC libraries

Now that you have configured your project and referenced the FrameAC libraries, you can use the Visual Basic Object Browser to look at the contents of these libraries. Open the browser and take a moment to familiarise yourself with their contents.

To open the Object Browser, press CTRL+ALT+J, or choose View > Object Browser. The object browser appears. On the left hand pane will be a list with all the names of the libraries currently referenced.

Important: All the FrameAC libraries will be prefixed with *Interop.*

For more information please look at “FACLib” on page 21

Writing the Code

The following code creates a basic plug-in that does the following:

- Connects to the FrameMaker session when FrameMajer starts up
- Posts a menu and commands to the FrameMaker menu bar
- Responds to the menu commands

clsNetBasicTutorial.vb Listing

Following is a code listing for the .Net basic tutorial plug-in. After the listing you will find line-by-line explanations of the code.

```
Imports FACLib
Imports FACLib.FDK_ObjectDefs
Imports FACLib.FDK_Flags
Imports System.Runtime.InteropServices
<System.Runtime.InteropServices.ProgId("NetBasicTutorial.clsNetBasicTutorial")> _
Public Class clsNetBasicTutorial
Private Const HELLO_WORLD_CMD As Long = 921
Private Const GOODBYE_CRUEL_WORLD_CMD As Long = 922
Private WithEvents theSession As FACLib.FMSession
```

```
Public Sub Initialise(ByVal iSession As Object, ByVal cmdFirst As Integer)
' store the passed iSession to local global variable
Me.theSession = iSession
' First Authorise the FrameAC connection - do nothing on failure
Dim FACAuthorise As New FRAMEACLib.FrameAuthorise
If Me.theSession.Initialise(FACAuthorise) = 1 Then
 AddMenus()
End If
End Sub
```

```
Private Sub AddMenus()
Dim oMainMenu As FMMenu
Dim oMenu As FMMenu
Dim oNewCommand As FACLib.FMCommand
' get the FrameMaker main menu
oMainMenu = Me.theSession.ConvertToFMMenu _
(Me.theSession.GetNamedObject("!MakerMainMenu", mFO_Menu, Nothing))
' check if the new menu already exists
oMenu = Me.theSession.GetNamedObject("NetBasicTutorialMENU", mFO_Menu, Nothing)
If oMenu Is Nothing Then
 ' menu doesn't exist so now add it
 oMenu = Me.theSession.DefineMenu("NetBasicTutorialMENU", "NetBasicTutorial")
 ' add the menu to the framemaker main menu
```

```
 oMainMenu.AddMenuToMenu(oMenu)
 ' now that we have the menu we can add commands to it
 ' create the command object
 oNewCommand = Me.theSession.DefineCommand(HELLO_WORLD_CMD, _
 "EventResponderHelloWorld", "Hello World", "")
 ' add the command to the menu
 oNewCommand.AddCommandToMenu(oMenu)

 oNewCommand = Me.theSession.DefineCommand(GOODBYE_CRUEL_WORLD_CMD, _
 "EventResponderGoodbyeWorldC", "Goodbye", "")
 oNewCommand.AddCommandToMenu(oMenu)
End If
End Sub
```

```
Private Sub theSession_OnApiCommand(ByVal command As Integer) _
Handles theSession.OnApiCommand
' branch the processing commands based on command
Select Case command
 Case Me.HELLO_WORLD_CMD
 Me.theSession.Alert("Hello World", mFF_ALERT_CONTINUE_NOTE)
 Case Me.GOODBYE_CRUEL_WORLD_CMD
 Me.theSession.Alert("Goodbye Cruel World", mFF_ALERT_CONTINUE_NOTE)
End Select
End Sub
End Class
```

Import Declarations

Lines Through are Import declarations. An import declaration imports namespace names from referenced projects and assemblies. These are useful as it means that any objects that are used do not have to be reference by the full object path. i.e. we can use line `mFO_Menu` instead of `FACLib.FDK_ObjectDefs.mFO_Menu` thus making the code less cumbersome.

COM Interop Declaration

Line is the COM Interop declaration this identifies the class as being available for COM and it also provides the name that it is registered as in COM. In this case it is `NetBasicTutorial.clsNetBasicTutorial` which is also the name that should be used to register the plugin for FrameMaker.

Class Declaration

Line is the standard method for declartion of class in Visual Basic .Net. All classes need a similar line unlike VB6 where the name of the class was taken from the property of the class.

General Declarations

Lines to are general declarations. The constants are useful for identifying the menu commands, which are used later in sections of the code, but are only available within the class (Private). The FrameAC DLL marshalls the events and notifications for all FrameAC plug-ins and controller applications. For this reason, the command numbers you use must be unique for all commands in all FrameAC applications. In this tutorial, the commands begin at an arbitrarily high number — 921.

Line declares a private global variable for the session that will be used throughout the code once initialisation is accomplished. This variable has been declared `withevents` which allows us to trap the events that occur on the session.

Initialisation

Lines through comprise the initialisation method on the clsNetBasicTutorial class. Every plug-in must have this method. It is called by FrameAC when FrameMaker starts up and begins the init process for every FrameAC plug-in that is registered in maker.ini. FrameAC calls this method to establish the plug-in's connection with the FrameMaker session.

Line sets the session to the global variable.

Lines and authorise and establish the connection, tesing to make sure the connection is valid, and line calls a routine to post the menus.

Adding Menus

Lines through define Sub routine that posts the menus to FrameMaker. These actions could be performed in the initialisation routine, but they are performed here to better organise the code.

Lines and create an object that represents the FrameMaker main menu. Note that the menu name is reserved within FrameMaker. For a complete listing of reserved menu and command names, see the FrameMaker online documentation for customising the Framemaker user interface on Windows, or see the `menu.cfg` files stored in `..\fminit\fmstruct\` and `...\fminit\maker\` directories.

> **Important:** Explicit/Implicit cast of the FrameMaker main menu object is not available and we must therefore use a method on the FMSession to convert the returned object to the desired object type. see "FMSession" on page 195

Line tries to get an object for the menu we will create. If the object is Nothing, then the menu doesn't exist, and so the code creates it.

Lines through create the menu and the commands, and adds them to the main menu. When creating menus and commands, the first string in the method is for the item's internal name. These names should be unique among all FrameAC applications. The second string is the label that appears in the menu; these do not have to be the same as the item name, and you can have more than one command or menu with the same label. Commands have a third string to represent the keyboard shortcut. These must be unique for all FrameAC applications.

Notice that the commands are created with the constants we defined in lines and . When the user chooses one of the commands, FrameAC will pass that constant to the command event.

Responding to Menu Commands

Lines through is the method that handles the FMSession.OnApiCommand event. This event is triggered by FrameAC when a user chooses one of the commands that we added to the menus. This method uses a Select Case block to test which command the user chose, and branch off to the appropriate code. In this tutorial, the commands determine which message to post in the alert box.

Registering the Plug-In

see "Registering the Plug-in" on page 25

Debugging the Plug-in

Before running final build on your plug-in and delivering it, you should try debugging it using the Visual Basic .Net IDE. When you run it this way, if there are any situations where objects are not set or other problems arise in the code, VB .Net notifies you of the problem and will highlight the offending line of code.

To set up debugging for a plug-in, you specify what product will load the plug-in, then run it from VB.Net, as follows:

1. Choose Project > Properties

   This will open the properties dialog

2. Choose Configuration Properties > Debugging

   Navigate to the debugging properties

3. Select the *Start external program* option

4. Either type in the full path to the FrameMaker.exe, or browse to the exe.

5. Click OK

6. Chose Debug > Run or Press F5

   VB.Net starts up a FrameMaker instance, which loads the plug-in on initialisation.

   From there you can go to the FrameMaker application window and try out the commands.

Look for a menu name NetBasicTutorial, with the commands Hello and Goodbye.

If you are fortunate, everything will work with no problems. If there is a problem, VB.Net posts an alert with options for you to proceed. For more information about debugging VB.Net, see your VB.Net documentation.

7. Close Framemaker to end the debugging session.

Final Installation

see "Final Installation" on page 26

Tutorial: Creating a FrameAC Controller App in Visual Basic .Net

This tutorial leads you through the steps to create a controller application. These steps include:

- Creating and configuring a new VB.Net project
- Writing the code
- Registering controller applications
- Debugging the controller application
- Final Installation

You will create this plug-in in Visual Basic .Net. The code uses events, methods and properties that may not be familiar to you. For more information about specific events, methods and properties, please refer to the appropriate chapters of this manual.

Create a New Project

FrameAC controller applications are standard EXE applications that stand alone.

To create and configure the new project:

1. Create the new Project

   Start Visual Studio .Net and create a new Visual Basic Windows Application.

   Call the new application `NetBasicController`

   VB.Net will create a blank form on creation of new windows application. Leave that form open in your work space.

2. Add References to the FrameAC libraries

   Choose Project > Add Reference to open the References dialog box.

   Select the COM Tab

   Then find the following entries and click select

   - Mekon FrameAC Type Library
   - Mekon FrameAC Library

- Mekon FDK Types
- Mekon FrameAC UDT Type Library

Then click OK.

3. Specify the name and text of the controller's form

   With the form selected, in the Name field of the Properties window enter the name, `ControllerForm1`. In the Text field, enter the name `FrameAC .Net Controller`.

4. Add a button to the form.

   Draw a button on the form. With the button selected, in the Name field of the Properties window enter the name, `TakeControl`. In the text field of the properties window enter the caption, `TakeControl`

5. Double-click the button to open the code window. You should see an empty Sub routine for `TakeControl_Click`.

6. Add the following code to the Sub routine created above.

```
Dim oDoc As FMDocument
    Me.theSession = Me.theEx.Session
    If Me.theSession Is Nothing Then
        MsgBox("Please Start FramerMaker!")
        Exit Sub
    End If
    If Not Me.theSession.Initialise(Me.authorisation) = 1 Then
        MsgBox("Failed to connect to FrameMaker!")
        Exit Sub
    End If
    oDoc = Me.theSession.ActiveDoc
    If oDoc Is Nothing Then
        Me.theSession.Alert("There are no active documents", _
        mFF_ALERT_CONTINUE_WARN)
    Else
        Me.theSession.Alert(oDoc.name & " is currently active", _
        mFF_ALERT_CONTINUE_NOTE)
        oDoc.AddText(oDoc.TextSelection.beg, "Hello World!")
      End If
```

7. Save the project.

   You should save the project with a name and location that makes sense to you.

The project is now configured to have access to all the FrameAC events, methods, objects and data types. You have also provided it with a form that you will use to initiate the routines and connect with FrameMaker and perform actions in a FrameMaker document. Note for this example FrameMaker must already be open prior to running the application.

Writing the Code

The following code creates a controller application that does the following:

- Connects to an existing FrameMaker session when application starts
- Verifies that there is an open/active document in FrameMaker
- Adds some text to the document

Form1.vb listing

Following is the code listing for the .Net basic tutorial controller application. After the listing you will find a line-by-line explanation of the code.

> **Important:** The `Windows Form Designer generated code` that is automatically generated for the form objects has been ommited to limit the listing.

```
Imports FRAMEACLib
Imports FACLib
Imports FACLib.FDK_Flags
Public Class ControllerForm1
     Inherits System.Windows.Forms.Form
     Private authorisation As New FrameAuthorise
     Private theEx As New FrameEx
     Private theSession As FMSession
#Region " Windows Form Designer generated code "
#End Region
     Private Sub TakeControl_Click _
     (ByVal sender As System.Object, ByVal e As System.EventArgs) _
     Handles TakeControl.Click
          Dim oDoc As FMDocument
          Me.theSession = Me.theEx.Session
          If Me.theSession Is Nothing Then
               MsgBox("Please Start FramerMaker!")
               Exit Sub
          End If
          If Not Me.theSession.Initialise(Me.authorisation) = 1 Then
```

```
                MsgBox("Failed to connect to FrameMaker!")
                Exit Sub
            End If
            oDoc = Me.theSession.ActiveDoc
            If oDoc Is Nothing Then
                Me.theSession.Alert("There are no active documents", _
                mFF_ALERT_CONTINUE_WARN)
            Else
                Me.theSession.Alert(oDoc.name & " is currently active", _
                mFF_ALERT_CONTINUE_NOTE)
                oDoc.AddText(oDoc.TextSelection.beg, "Hello World!")
            End If
        End Sub
End Class
```

Import Declarations
Lines through are import declarations. An import declaration imports namespace names from reference projects and assemblies. These are useful as it means that any objects that are used don't have to be referenced by the full object path. i.e. we can use line `mFO_Menu` instead of `FACLib.FDK_ObjectDefs.mFO_Menu` thus making the code less cumbersome.

Form Declaration
Lines and are the form declarations. All forms in VB.Net are classes that inherit from the base form class.

General Declarations
Lines through are general declarations. These declare global private variables for the FrameMaker session and other objects necessary to establish the connection with the FrameMaker session.

Omitted Code Region
Lines and are the start and end points of the code that was omitted. This code is automatically generated for the form and does not have a direct bearing on our description of the interaction of the example.

Handling the Form Event
Lines through code the whole of the application's behaviour. After declaring a variable for a document object on line , the code established a connection through the FrameAC DLL's FrameEx object on line .

Lines through test that a FrameMaker instance has been started, and if not they post an alert and exit the subroutine.

Lines through test the connection, and if it is not valid they post an alert and exit the sub routine.

Line gets the active document object.

Lines through test to see that there is an active document, and if not they post an alert. If there is an active document then they post an alert giving the name of the active document (line).

Line adds the text "Hello world!" to the beginning position of the current text selection in the document.

Registering the Application

see “Registering the Plug-in” on page 25

Debugging the Application

Before running the final build on you controller application and delivering it, you should try running it through the VB.Net environment to debug it. This will highlight any situation where objects are not set or any other problems in the code.

The only requirement to run the controller application is that FrameMaker is already running before you start it. To run the application within the VB.Net Environment:

1. Start FrameMaker

   To test this application, you should open a document and make sure you have a selection or insertion point in it.

2. Choose Debug > Start (or press F5)

   VB.Net runs the program and displays the application’s form.

3. Click the Take Control button

   If all goes well, you should see an alert box that displays the document’s full path.

4. Dismiss the alert box

   Now your controller application should write "Hello World!" text into the active document to the left of the current selection or insertion point.

5. Break for debugging.

   Close the application form to break from debugging.

Final Installation

see “Final Installation” on page 26

Tutorial: Creating a FrameAC Plug-in for Visual Basic 6

This tutorial leads you through the steps to create a basic FrameAC plugin in Visual Basic 6. These steps include:

- Creating a new a VB project and designing the application's form
- Writing the code
- Registering the plug-in
- Debugging the plug-in
- Final installation

You will create this plug-in in Visual Basic. The code uses events, methods, and properties that may not be familiar to you. For more information about specific events, methods, and properties, please refer to the appropriate chapters of this manual.

Creating and Configuring a New Project

FrameAC plug-ins are ActiveX components, and so they must be contained in an ActiveX server. This server can be either a DLL or an EXE.

To create and configure the new project:

1. Create the new project.

   Start Visual Basic and create a new ActiveX project. Note that the project *must* be ActiveX and not a standard EXE.

2. Establish references to the FrameAC libraries.

   Choose Project > References to open the References dialog box. Then find the following entries and turn on their check boxes:

   - Mekon FrameAC Type Library
   - Mekon FrameAC Library
   - Mekon FDK Types
   - Mekon FrameAC UDT Type Library

   Then click OK.

   Note: ensure that the libraries are referenced in the above order.

3. Specify the name of the project.

   Choose Project > Properties to display the Project Properties dialog box. In the General tab, find the Project Name text box and enter `BasicTutorial`.

4. Specify the name of the plug-in class.

   In the Name field of the Properties window, enter the name, `clsBasicTutorial`.

5. Save the project.

   You should save the project with a name and location that makes sense to you.

The project is now configured to have access to all the FrameAC events, methods, objects, and data types. You have also provided it with the name you will use to register the plug-in in the .ini file.

Browsing the FrameAC libraries

Now that you have configured your project and referenced the FrameAC libraries, you can use the Visual Basic Object Browser to look at the contents of these libraries. Open the browser and take a moment to familiarise yourself with their contents.

To open the Object Browser, press F2, or choose View > Object Browser. The object browser appears. In the top-left corner of the browser is a drop-down list with names of all the libraries currently referenced, plus an entry for <All Libraries>. You choose which library to browse from that list.

FACLib

Choose FACLib from the drop-down list to view the objects and data types defined for FrameAC. Of particular interest are the constant numeric values:

- FDK_Errors — Error codes that describe the cause of an error state. Many methods can return a subset of these, or set these to the session property, `FA_errno`.
- FDK_Flags — Constant numeric values used to pass instructions to various methods.
- FDK_Notifications — Constants that identify what user-events have occurred. You use these in conjunction with the OnApiNotify event.
- FDK_OperationProperties — Constants you use to script the behaviour of methods to import, open, save, or update files.
- FDK_Properties — A listing of all the properties that can exist in all the FrameAC objects. Each object shows a subset of this list.
- FDK_TextItemFlags, FDK_TextItemTypes, and FDK_TextItemTypes2 — Constants used to control how you get text from a document.
- FDK_ValTypes — Constants to identify the different types of data in FrameAC.
- FDK_Values — Various enumerated values that are used for different object properties.

Following the list of constants are the various FrameAC objects. You should look at a few of them to see how they're organized. The objects are documented in Chapter 3, "Object Reference," with explanations of the various properties and links to the documentation for each method.

All this may seem daunting, but don't worry, you never need to memorise all this information and you don't need to know too much about it to continue with the tutorial. Your experience with FrameMaker and Visual Basic should help you fill in the gaps as you progress.

Writing the Code

The following code creates a basic plug-in that does the following:

- Connects to the FrameMaker session when FrameMaker starts up
- Posts a menu and commands to the FrameMaker menu bar
- Responds to the menu commands

BasicTutorial Listing

Following is a code listing for the basic tutorial plug-in. After the listing you will find line-by-line explanations of the code.

```
Option Explicit
Const HELLO_WORLD_CMD As Long = 901
Const GOODBYE_CRUEL_WORLD_CMD As Long = 902
Public WithEvents theSession As FMSession
```

```
Public Sub Initialise(iSession As Object, cmdFirst As Long)
' First authorise the FrameAC connection - do nothing on failure
Dim FACAuthorise As FrameAuthorise
     Set FACAuthorise = New FrameAuthorise
     If iSession.Initialise(FACAuthorise) = 1 Then
          Set theSession = iSession
          AddMenus
     End If
End Sub
```

```
Public Sub Destroy()
' Perform any necessary cleanup here
End Sub
```

```
Private Sub AddMenus()
Dim oMainMenu As FMMenu, oMMenu As FMMenu
Dim oNewCommand As FMCommand
     ' Get the framemaker main menu
     Set oMainMenu = theSession.GetNamedObject _
  ("!MakerMainMenu", mFO_Menu, Nothing)
```

```
     ' Check if the new menu already exists
     Set oMMenu = theSession.GetNamedObject("BasicTutorial", mFO_Menu, Nothing)
     If oMMenu Is Nothing Then
          ' menu doesn't exist so now add it
          Set oMMenu = theSession.DefineMenu _
("BasicTutorialMENU", "BasicTutorial")
          oMainMenu.AddMenuToMenu oMMenu

          ' Now that we have the menu we can add commands to it
          ' create the command object
          Set oNewCommand = theSession.DefineCommand _
 (HELLO_WORLD_CMD, "EventResponderHelloWorld", "Hello World", "")
          ' add the command to the menu
          oNewCommand.AddCommandToMenu oMMenu

          Set oNewCommand = theSession.DefineCommand _
 (GOODBYE_CRUEL_WORLD_CMD, "EventResponderGoodbyeWorldC", "Goodbye", "")
          oNewCommand.AddCommandToMenu oMMenu
     End If
End Sub
```

```
Private Sub theSession_OnApiCommand(ByVal cmd As Long)

     ' Branch the processing commands based on cmd
     Select Case cmd
          Case HELLO_WORLD_CMD
               theSession.Alert "Hello World", mFF_ALERT_CONTINUE_NOTE
          Case GOODBYE_CRUEL_WORLD_CMD
               theSession.Alert "Goodbye Cruel World", mFF_ALERT_CONTINUE_NOTE
     End Select
End Sub
```

General Declarations

Lines through are general declarations. The constants are useful for identifying the menu commands, which are used in later sections of the code. The FrameAC DLL marshalls the events and notifications for all FrameAC plug-ins and controller applications. For this reason,

the command numbers you use must be unique for all commands in all FrameAC applications. In this tutorial, the commands begin at an arbitrarily high number — 901.

Line declares a global variable for the session that will be used throughout the code once initialisation is accomplished.

Initialisation

Lines through comprise the initialisation Sub routine. Every plug-in must have this Sub routine. It is called by FrameAC when FrameMaker starts up and begins the init process — for every FrameAC plug-in that is registered in maker.ini, FrameAC calls this routine to establish the plug-in's connection with the FrameMaker session.

Lines through authorise and establish the connection, testing to make sure the connection is valid. Line sets the session to the global variable, and line calls a routine to post the menus.

Destruction

Lines through define the destruction routine. This Sub routine is also required for every FrameAC plug-in. This is a good opportunity to clean up any processes or temporary files that may remain when FrameMaker quits.

Adding Menus

Lines through are a Sub routine that posts the menus to FrameMaker. These actions could be performed in the initialisation routine, but they are performed here to better organise the code.

Lines and create an object that represents the FrameMaker main menu. Note that the menu name is reserved within FrameMaker. For a complete listing of reserved menu and command names, see the FrameMaker online documentation for customising the FrameMaker user interface on Windows, or see the `menus.cfg` files stored in the `...\fminit\fmstruct\` and `...\fminit\maker\` directories.

Lines tries to get an object for the menu we will create. If the object is Nothing, then the menu does not exist, and so the code creates it.

Lines through create the menu and the commands, and adds them to the main menu. When creating menus and commands, the first string in the method is for the item's internal name. These names should be unique among all FrameAC applications. The second string is the label that appears in the menu — these do not have to be the same as the item name, and you can have more than one command or menu with the same label. Commands have a third string to represent the keyboard shortcut. These must be unique for all FrameAC applications.

Notice that the commands are created with the constants we defined in Lines and . When the user chooses one of these commands, FrameAC will pass that constant to the command event.

Responding to Menu Commands

Lines through define the OnApiCommand event. This event uses a Select Case block to identify which command the user chose, and branch off to the appropriate code. In this tutorial, the commands determine which message to post in an alert box.

Registering the Plug-in

Before you can run or even dubug the plug-in, you must register it in the maker.ini file. While FrameMaker supports separate maker.ini files for different users, it's advisable to use the main file that is stored in the FrameMaker product directory. Open this file and look for the following sections:

- [APIClients]
- [FrameCOMClients]

[APIClients]

Go to the [APIClients] section and make sure it has the following entry:

```
FrameAC=Standard, Mekon FrameAC, <InstallDir>\Mekon\FrameAC\FRAMECOM.DLL,all
```

where `<InstallDir>` is the location of the FrameAC installation. If you do not find this entry, then you probably have a faulty installation of FrameAC. First make sure you have your licence numbers written down for safe-keeping. Then quit all your programs, and try to install FrameAC again.

[FrameCOMClients]

Go to the `[FrameCOMClients]` section of your maker.ini file. This is where you will register your FrameAC plug-in.

To register this tutorial plug-in, add the following entry:

```
NetBasicTutorial.clsNetBasicTutorial={xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}
```

This registers the plug-in so FrameAC can initialise it.

Debugging the Plug-in

Before running a final make on your plug-in and delivering it, you should try building it through the Visual Basic environment to debug it. When you run it this way, if there are any situations where objects are not set or other problems arise in the code, Visual Basic notifies you of the problem and highlights the offending lines of code.

To set up debugging for a plug-in, you specify what product will load the plug-in, then run it from Visual Basic, as follows:

1. Choose Project > Properties.

   In the Project dialog box, go to the Debugging tab.

2. Turn on the Start Program option

3. Either type the full path to the FrameMaker EXE , or browse to the EXE to enter the path in the text box. Then click OK

4. Choose Run > Start.

   Visual Basic starts up a FrameMaker process, which loads the plug-in on initialisation. From there you can go to the FrameMaker application window and try out the commands. Look for a menu name BasicTutorial, with the commands Hello and Goodbye.

   If you are fortunate, everything will work with no problems. If there is a problem, Visual Basic posts an alert with options for you to proceed. For more information about debugging in Visual Basic, see your Visual Basic documentation.

5. Break out of debugging.

   When you quit the FrameMaker process that started up in step 4, your plug-in may still be running. In that case, you won't be able to modify it. To shut down the plug-in, choose Run > End.

Final Installation

When you are sure the plug-in works as intended and is free of bugs, you can run a final make on it and deliver it to your users. All users of your plug-in must have FrameAC installed with at least a user's licence. The users must also have the following entry in the [FrameCOMClients] section of their maker.ini files:

```
TutorialBasic.clsTutorialBasic={xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}
```

To run a make on your plug-in, choose File > Make BasicTutorial.exe (or Make BasicTutorial.dll). This creates an EXE or dll that you can deliver to your users.

You must then register the exe or dll file on the target users' machine(s).

Tutorial: Creating a FrameAC Controller App.

This tutorial leads you through the steps to create a controller application using Visual Basic 6. These steps include:

- Creating and configuring a new VB project
- Writing the code
- Registering controller applications
- Debugging the controller application
- Final build

You will create this plug-in in Visual Basic. The code uses events, methods, and properties that may not be familiar to you. For more information about specific events, methods, and properties, please refer to the appropriate chapters of this manual.

Creating a New Project

FrameAC controller applications are standard EXE applications that stand alone.

To create and configure the new project:

1. Create the new project.

   Start Visual Basic and create a new standard EXE. When Visual Basic opens the new project, it creates a blank form. Leave that form open in your work space.

2. Establish references to the FrameAC libraries.

   Choose Project > References to open the References dialog box. Then find the following entries and turn on their check boxes:

   - Mekon FrameAC Type Library
   - Mekon FrameAC Library
   - Mekon FDK Types
   - Mekon FrameAC UDT Type Library

   Then click OK.

   Note: ensure that the libraries are referenced in the above order.

3. Specify the name of the project.

   Choose Project > Properties to display the Project Properties dialog box. In the General tab, find the Project Name text box and enter `BasicController`.

4. Specify the name and caption of the controller's form.

   With the form selected, in the Name field of the Properties window enter the name, `ControllerForm1`. In the Caption field, enter the name `FrameAC Controller`.

5. Add a button to the form.

   Draw a button on the form. With the button selected, in the Name field of the Properties window enter the name, `TakeControl`. In the Caption field of the properties window enter the caption, `TakeControl`.

6. Double-click the button to open the code window. You should see an empty Sub routine for `TakeControl_Click()`.

7. Save the project.

   You should save the project with a name and location that makes sense to you.

The project is now configured to have access to all the FrameAC events, methods, objects, and data types. You have also provided it with a form that you will use to initiate the routines that connect with FrameMaker and perform actions in FrameMaker documents.

Writing the Code

The following code creates a basic plug-in that does the following:

- Connects to the FrameMaker session when FrameMaker starts up
- Posts a menu and commands to the FrameMaker menu bar
- Responds to the menu commands

BasicTutorial Listing

Following is a code listing for the basic tutorial controller application. After the listing you will find line-by-line explanations of the code.

```
Option Explicit

Public authorisation As New FrameAuthorise

Public theEx As New FrameEx

Public theSession As FMSession
```

```
Private Sub TakeControl_Click()

Dim oDoc As FMDocument


     Set theSession = theEx.Session

     If Not theSession.Initialise(authorisation) = 1 Then

          MsgBox "Failed to connect with FrameMaker!"

          Exit Sub

     End If


     Set oDoc = theSession.ActiveDoc

     theSession.Alert oDoc.Name + " is currently active", _

               mFF_ALERT_CONTINUE_NOTE


     oDoc.AddText oDoc.TextSelection.beg, "Hello World! "

End Sub
```

General Declarations

Lines through are general declarations. These declare global variables for the FrameMaker session and other objects necessary to establish the connection with the FrameMaker session..

Handling the Form Event

Lines through code the whole of this application's behavior. After declaring a variable for a document object on line , the code establishes a connection through the FrameAC DLL's FrameEx object on line .

Lines through test the connection, and if it is not valid they post an alert and exit the Sub routine. The remaining lines of code get the session's active document. Then they get the

document's name property, and insert text at the beginning position of the current text selection.

Registering the Application

Before you can run the application, you must register it in the maker.ini file. While FrameMaker supports separate maker.ini files for different users, it's advisable to use the main file that is stored in the FrameMaker product directory. Open this file and look for the following sections:

- [APIClients]
- [FrameCOMClients]

[APIClients]

Go to the [APIClients] section and make sure it has the following entry:

```
FrameAC=Standard, Mekon FrameAC, <InstallDir>\Mekon\FrameAC\FRAMECOM.DLL,all
```

where `<InstallDir>` is the location of the FrameAC installation. If you do not find this entry, then you probably have a faulty installation of FrameAC. First make sure you have your licence numbers written down for safe-keeping. Then quit all your programs, and try to install FrameAC again.

[FrameCOMClients]

Go to the `[FrameCOMClients]` section of your maker.ini file. You enter a line here that registers all FrameAC controller applications.

To register crontroller applications, add the following entry:

```
FrameAC.FrameEx={E6E0DCF4-A895-11D3-91B3-0050BAAE27FB}
```

Debugging the Plug-in

Before running a final make on your plug-in and delivering it, you should try running it through the Visual Basic environment to debug it. When you run it this way, if there are any situations where objects aren't set or other problems arise in the code, Visual Basic notifies you of the problem and highlights the offending lines of code.

The only requirement to run a controller application is that FrameMaker is already running before you start it. To run the application within the Visual Basic environment:

1. Start up FrameMaker

   To test this application, you should open a document and make sure you have a selection or insertion point in it.

2. Choose Run > Start

   Visual Basic runs the program and displays the application's form.

3. Click the TakeControl button

If all goes well, you will hear a system beep to accompany the alert box you programmed as the first action in your Sub routine. If FrameMaker isn't visible, move the FrameMaker application window to the front. You should see an alert box that displays the document's full path.

4. Dismiss the alert box.

   Now your controller application should write some text into the active document to the left of the current selection or insertion point.

5. Break out of debugging.

   There are two ways to break out of debugging:

   - Click the close box on the form — Visual Basic will return to the front of your desk top
   - Bring Visual Basic to the front of your desk top and choose Run > End

   You must end the application's process before you can edit the code any further.

6. Generate a failed state

   Go to the FrameMaker application and close all documents. Then choose Run > Start to run your controller application. This time you will get an error state. Click debug, and you will see that you have an error on line .

   You get this error because the `oDoc` object is set to `Nothing`, yet the string you pass to the Alert method depends on a valid document object. To gracefully handle this possibility, you should test `oDoc` before relying on it. You can either use the ObjectValid method, or test whether `oDoc` is set to `Nothing`.

Final Installation

When you are sure the plug-in works as intended and is free of bugs, you can run a final make on it and deliver it to your users. All users of your plug-in must have FrameAC installed with at least a user's license. The users must also have the following entry in the [FrameCOMClients] section of their maker.ini files:

```
FrameAC.FrameEx={E6E0DCF4-A895-11D3-91B3-0050BAAE27FB}
```

To run a make on your plug-in, choose File > Make BasicController.exe. This creates an EXE you can deliver to your users.

In addition, a deliverable EXE must include the required Visual Basic libraries. See your Visual Basic documentation for more information.

Tutorial: Creating FrameAC Scripts

FrameAC includes a scripting environment that you can use to create scripts that manipulate FrameMaker documents and objects. You can use these scripts to automate a wide range of tasks you perform manually. You can create the following types of scripts:

- Visual Basic scripts (VBS)

- JavaScript scripts
- Event-triggered scripts

This section of the tutorial describes how to create each type of script.

Visual Basic Script

The following script is a simple Hello World script that displays a message in an alert. These steps illustrate how to open the script editor, create a script, and run it.

To create the script:

1. Open the script editor.

   Choose ActiveScript > Script Editor — the script editor appears with Untitled1 as the current script.

2. Set up your scripting environment.

   To set up the environment:

   - Choose VBScript from the Language drop-down list
   - Click Font to set up the text display in the script editor.

3. Enter your code.

   This script is very simple. It only consists of the following line:

   ```
   Alert "Hello World", mFF_ALERT_CONTINUE_NOTE
   ```

4. Execute the script.

   Click the Execute button — an alert box should appear with the message, "Hello World".

5. Save the script

   Click Save Script — the Save As dialog box appears. Navigate to a location to save the script. Make sure the Save File Type is set to VB Script Files, then choose a name and save your script. Save the script as "HelloWorld.vbs". Ensure that you include the *.vbs*. This ensures that the script is saved as a VBScript.

6. Open the script you just saved.

   Notice that the current script is still Untitled1. Click Open Script, choose the script you just saved, then click Open.

The code for VB scripts corresponds directly with the descriptions in the reference chapters. However, the session object is implied, so you don't need to call session methods through it. For example, assume you create an application in Visual Basic, and create a session object named `mySession`. In that case, you would make a call to the Alert method through that session object — `mySession.Alert "Hello World", mFF_ALERT_CONTINUE_NOTE`. But in the code for this script, you did not need to create the session object, nor did you call the method through that object.

Another difference between Visual Basic and Visual Basic Script is that scripting syntax is slightly different, and VBS does not support everything you can do with Visual basic. For more information about these differences, see your documentation about the respective programming languages.

JavaScript

Creating a script using JavaScript is just as easy as creating one in VBS. You follow the same process except you:

- Choose JScript in the Langage drop-down list
- Save the script as a .js file.
- Write the code using valid JavaScript syntax

To create a Hello World script in JavaScript:

1. Go back to the Untitled1 script.

   Click the Untitled1 tab in the script editor.

2. Choose JScript in the Language drop-down list.
3. Change the code so it appears as follows:

   ```
   alert("Hello World", 2);
   ```

   Note the use of parentheses and the semi-colon to close the statement.

4. Run the script as you did for the VB script.
5. Save the script as a file with a .js extension.

Event-triggered Script

FrameAC scripting can respond to events that occur within a FrameMaker document or session. The Event Editor lists all the events a script can handle, and briefly describes each one. These events include notifications, which correspond to the notification points in the FrameAC OnApiNotify event (see Chapter 4, “Event Reference”).

To trigger a script from an event, you create the script and assign it to an event. This creates an event profile. Then you direct FrameAC to respond to that profile. To set up event handling:

1. Create a script.

   For this tutorial, you will use the Hello World script that you saved.

2. Open the Event Editor.

   Choose ActiveScript > Event Editor — the Event Editor window appears. Make sure the Notifications tab is selected. Notice that it lists a number of events, their associated operations, and descriptions of when each occurs within its operation.

3. Specify a profile name in the Profile name text box.

   Enter "Hello World Profile".

4. Provide a description of the profile in the Profile Description text box.

5. Assign a script to an event.

   Select the first event in the list — FA_Note_Profile_Applied. Then click Assign. In the dialog box that appears, navigate to the script file, HelloWorld.vbs, then click Open.

6. Save the profile.

   Click Save Profile. In the dialog box that appears, save the profile as HelloWorld.pro.

7. Click Close to close the Event Editor window.

8. Activate the loaded profile.

   Choose ActiveScript > Enable loaded event profile. As you do this, the HelloWorld.vbs script should execute, displaying an alert box.

You can create event profiles that respond to any number of events, all with different scripts. You should experiment with this so you can see how often these events occur in your normal work flow.

Tutorial: Calling script from Visual Basic

You can call a saved script from a FrameAC plugin to perform the specified tasks. One use for this might be to provide general users a way to customize your plugin without having to change the VisualBasic code. To call a script from VisualBasic, you must prepare the VisualBasic code to recognise the script, and then make the call to execute it.

Preparing the Visual Basic module

Before you can call a script from Visual Basic, you must make the following additions to your Visual Basic module.

1. Add two more constants to the module as follows:

   Follow the instructions as per Tutorial: Creating a FrameAC Plug-in.

```
Const EXEC_SCRIPT_CMD As Long = 903
Const EXEC_SCRIPT_PARAM_CMD As Long = 904
```

2. In the AddMenus procedure add the following code:

```
' Show how scripts can be used
' create the command object for executing sciptsSet
oNewCommand = theSession.DefineCommand(EXEC_SCRIPT_CMD, "Execute Script", "Execute
Script", "")

' add the scripting command to the menu
oNewCommand.AddCommandToMenu oMMenu
```

```
' create the command object for executing scipts with parametersSet
oNewCommand = theSession.DefineCommand(EXEC_SCRIPT_PARAM_CMD, "Execute Script With
Parameters", "Execute Script With Parameters", "")

' add the scripts with parameters command to the menu
oNewCommand.AddCommandToMenu oMMenu
```

Executing the script

To execute the script from your Visual Basic module:

1. In the theSession_OnApiCommand procedure add the following code:

```
Dim retval As Integer
Dim ScriptText As String
Dim testString As String
```

2. In the theSession_OnApiCommand procedure add the following code to the case statement:

For Javascript

```
Case EXEC_SCRIPT_CMD
ScriptText = "alert (" + chr(34) + "This was executed by the script" + chr(34)
+ ", 2);"
retval = theSession.ExecuteScript(ScriptText, ast_TEXT_JSCRIPT)
```

For VB Script

```
ScriptText = "Alert " + Chr(34) + "This was executed by the script" + Chr(34) +
", 0"
retval = theSession.ExecuteScript(ScriptText, ast_TEXT_VBSCRIPT)
```

Executing the script with parameters

To execute the script with parameters:

1. Add a new class to your project called clsData and add the following code to it:

```
Option Explicit

Public m_Archive_ID As Long
Public m_Doc_Date As String
Public m_Doc_Pages As Integer
Public m_Doc_Path As String
Public m_Doc_Subject As String
```

```
'*********************************************************************
' Function : Class_Initialize
' Purpose  : Set up the members
'*********************************************************************
Private Sub Class_Initialize()
' Clear all values
ResetMembers
End Sub

'*********************************************************************
' Function : ResetMembers
' Purpose  : Initialise the members
'*********************************************************************
Public Function ResetMembers()
m_Archive_ID = 0
m_Doc_Date = ""
m_Doc_Pages = 0
m_Doc_Path = ""
m_Doc_Subject = ""
End Function

'*********************************************************************
' Function : SetMembers
' Purpose  : Give the members a value
'*********************************************************************
Public Function SetMembers()
m_Archive_ID = 1000
m_Doc_Date = "01-01-2005"
m_Doc_Pages = 52
m_Doc_Path = "C:\Temp\Test"
m_Doc_Subject = "Testing scripting"
End Function
```

2. In the `Session_OnApiCommand` procedure add the following code to the case statement:

For Javascript

```
Case EXEC_SCRIPT_PARAM_CMD
Dim clsTest As New clsData

'########################################
```

```
'# Please remember when using jscript, it is case sensitive - the script variable
'# is g_Param
'########################################
ScriptText = "g_Param.SetMembers();"
retval = theSession.ExecuteScriptWithParam(ScriptText, ast_TEXT_JSCRIPT, clsTest)


testString = "Class members are: " + vbCrLf
testString = testString + "Archive ID: " + CStr(clsTest.m_Archive_ID) + vbCrLf
testString = testString + "Doc Date: " + clsTest.m_Doc_Date + vbCrLf
testString = testString + "Doc Pages: " + CStr(clsTest.m_Doc_Pages) + vbCrLf
testString = testString + "Doc Path: " + clsTest.m_Doc_Path + vbCrLf
testString = testString + "Doc Subject: " + clsTest.m_Doc_Subject + vbCrLf


MsgBox testString, vbMsgBoxSetForeground + vbInformation + vbOKOnly, "Script
execution result"
```

For VB Script

```
Case EXEC_SCRIPT_PARAM_CMD
Dim clsTest As New clsData


ScriptText = "g_param.SetMembers()"
retval = theSession.ExecuteScriptWithParam(ScriptText, ast_TEXT_VBSCRIPT, clsTest)


testString = "Class members are: " + vbCrLf
testString = testString + "Archive ID: " + CStr(clsTest.m_Archive_ID) + vbCrLf
testString = testString + "Doc Date: " + clsTest.m_Doc_Date + vbCrLf
testString = testString + "Doc Pages: " + CStr(clsTest.m_Doc_Pages) + vbCrLf
testString = testString + "Doc Path: " + clsTest.m_Doc_Path + vbCrLf
testString = testString + "Doc Subject: " + clsTest.m_Doc_Subject + vbCrLf


MsgBox testString, vbMsgBoxSetForeground + vbInformation + vbOKOnly, "Script
execution result"
```

3 Object Reference

FrameAC represents all the objects in a FrameMaker document, book, or session as VisualBasic objects. These FrameAC objects represent the data via properties, and can be manipulated via property settings and methods. Every FrameAC object is derived from the generic object, FMObject. This means that every FrameAC object inherits the properties and methods of FMObject.

To begin processing FrameMaker files and data, you must enter through the current FMSession object. The same FMSession is current and active from the moment FrameMaker starts to the moment FrameMaker exits. From the FMSession object you can do things such as initialise your plug-in menus, open documents, get a list of open documents, get the currently active document, close documents, and quit the FrameMaker session.

Most objects in a FrameMaker document have a UID property. The UID is an integer that uniquely identifies it among all other objects of the same type. The UID is saved with the document—UIDs are persistent across FrameMaker sessions.

Following is a reference of all the FrameAC objects, their properties, and their methods.

FMAnchoredFrame

Details

An anchored frame is a graphic object that is tied to a specific location in the text. The text that contains the anchor must be within a flow. An anchored frame can contain any graphic objects except other anchored frames. (To nest one anchored frame within another, you must have a text frame within the parent anchored frame, and then anchor the child to the text in that text frame.)

For different anchored frame types (AnchorType property), certain other properties of the FMAnchoredFrame object must have specific values. For example, if the AnchorType is mFV_ANCHOR_RUN_INTO_PARAGRAPH, the SideOffset must be 0. The following table lists the constraints on properties for different anchored frame types:

AnchorType value	Property constraints
mFV_ANCHOR_INLINE	SideOffset = 0
mFV_ANCHOR_TOP	SideOffset = 0
mFV_ANCHOR_BELOW	BaselineOffset = 0
mFV_ANCHOR_BOTTOM	

AnchorType value	Property constraints
mFV_ANCHOR_RUN_INTO_PARAGRAPH	SideOffset = 0 BaselineOffset = 0 AFrameIsFloating = 0 AFrameIsCropped = 0
mFV_ANCHOR_SUBCOL_FARTHEST mFV_ANCHOR_SUBCOL_INSIDE mFV_ANCHOR_SUBCOL_LEFT mFV_ANCHOR_SUBCOL_NEAREST mFV_ANCHOR_SUBCOL_OUTSIDE mFV_ANCHOR_SUBCOL_RIGHT mFV_ANCHOR_TEXTFRAME_FARTHEST mFV_ANCHOR_TEXTFRAME_INSIDE mFV_ANCHOR_TEXTFRAME_LEFT mFV_ANCHOR_TEXTFRAME_NEAREST mFV_ANCHOR_TEXTFRAME_OUTSIDE mFV_ANCHOR_TEXTFRAME_RIGHT	AFrameIsFloating = 0 AFrameIsCropped = 0

See also

“FMObject” on page 166, “FMGraphic” on page 155, “FMFlow” on page 141

Properties

Property:	Type:	Description:
AFrameIsCropped	F_Int	True if Cropped is enabled
AFrameIsFloating	F_Int	True if Floating is enabled
Alignment	F_Int	Type of alignment: mFV_ALIGN_CENTER mFV_ALIGN_INSIDE mFV_ALIGN_OUTSIDE mFV_ALIGN_LEFT mFV_ALIGN_RIGHT
AnchorType	F_Int	Where frame is anchored: mFV_ANCHOR_BELOW mFV_ANCHOR_BOTTOM mFV_ANCHOR_INLINE mFV_ANCHOR_RUN_INTO_PARAGRAPH mFV_ANCHOR_SUBCOL_FARTHEST mFV_ANCHOR_SUBCOL_INSIDE mFV_ANCHOR_SUBCOL_LEFT mFV_ANCHOR_SUBCOL_NEAREST mFV_ANCHOR_SUBCOL_OUTSIDE mFV_ANCHOR_SUBCOL_RIGHT mFV_ANCHOR_TEXTFRAME_FARTHEST mFV_ANCHOR_TEXTFRAME_INSIDE mFV_ANCHOR_TEXTFRAME_LEFT mFV_ANCHOR_TEXTFRAME_NEAREST mFV_ANCHOR_TEXTFRAME_OUTSIDE mFV_ANCHOR_TEXTFRAME_RIGHT mFV_ANCHOR_TOP
BaselineOffset	F_Metric	Baseline offset
Element	FMElement	If the anchored frame is in a structured flow in a FrameMaker+SGML document, the element object containing the anchored frame
FirstGraphicInFrame	FMGraphic	First graphic object in frame

Property:	Type:	Description:
InTextFrame	FMTextFrame	FMTextFrame in which anchored frame appears
InTextObj	FMObject	FMSubCol or FMTextFrame in which anchored frame appears
LastGraphicInFrame	FMGraphic	Last object in frame
NextAFrame	FMAnchoredFrame	Next FMAnchoredFrame object in the current text frame
PrevAFrame	FMAnchoredFrame	Previous FMAnchoredFrame object in the current text frame
SideOffset	F_Metric	Near side offset
TextLoc	udtTextLoc	Text location of the anchor symbol

FMArc

Details

FMArc objects represent arcs that are drawn with the FrameMaker drawing tools.

See also

"FMObject" on page 166, "FMGraphic" on page 155

Properties

Property:	Type:	Description:
DTheta	F_Metric	Arc angle length (–360 degrees to 360 degrees)
Theta	F_Metric	Start angle (0 degrees to 360 degrees)

FMAttrCondExpr

Details

FMAttrCondExpr objects represent conditional attribute expressions.

See also

"FMObject" on page 166

Properties

Property:	Type:	Description:
NextAttrCondExprInDoc	FMAttrCondExpr	ID of the next FMAttrCondExpr in document
AttrCondExprStr	F_String	Conditional expression
AttrCondExprIsActive	F_Int	True if the conditional expression is active

FMBodyPage

Details

FMBodyPage objects represent the body pages in a document. A body page is a FrameMaker document page that contains the flow of body text. Other page types include master pages, reference pages, and hidden pages.

A body page has a PageBackground property, which determines the master page used to describe the layout of the body page. However, for double-sided documents (that use Right and Left page layouts) the default PageBackground doesn't indicate whether the page is Right or Left. To determine if a body page has a Left or a Right master page background when its PageBackground property is set to mFV_BGD_DEFAULT, query its PageIsRecto property.

See also

"FMObject" on page 166, "FMPage" on page 168, "FMHiddenPage" on page 159, "FMMasterPage" on page 163, "FMReferencePage" on page 191

Properties

Property:	Type:	Description:
MasterPage	F_String	Name of master page background for body page if PageBackground is set to mFV_BGD_OTHER. It is NULL if PageBackground is set to mFV_BGD_DEFAULT or mFV_BGD_NONE.
PageBackground	F_Int	Type of master page background: mFV_BGD_DEFAULT: The page has a Left or Right master page background if the document is double-sided, or a Right master page background if the document is single-sided. mFV_BGD_NONE: The page has no master page background. mFV_BGD_OTHER: The page has the custom master page background specified by MasterPage.
PageIsRecto	F_Int	True if right body page or False if left body page.
PageNum	F_Int	Page number.
PageNumString	F_String	Page number string.
PointPageNum	F_Int	Number of point page.
TheMasterPage	FMMasterPage	The master page object that pertains to the current body page.

FMBook

Details

FMBook objects represent the FrameMaker books that are open in a session. A FrameMaker book is a collection of document files. FrameMaker uses books to manage processing of the collection of documents such as numbering, generating TOCs and indexes, and otherwise managing the collection of documents as a single volume or publication. The FMBook object contains a collection of FMBookComponent objects. Each FMBookComponent indicates a specific FMDocument object.

FrameMaker supports XMP Metadata, which is a protocol to store information about a file as encoded packets that are available to external applications. For more information about XMP data, see “XMP Data in FrameMaker Documents” on page 419 of Appendix B, “FrameMaker Document and Session Architecture.”

See also

“FMObject” on page 166, “FMBookComponent” on page 58, and “FMDocument” on page 88

Properties

This reference manual divides the FMBook properties into the following groups:

- “Book General Properties” on page 43
- “Book PDF Properties” on page 45
- “Book Print Properties” on page 50
- “Book Structure Properties” on page 52
- “Book View Only Properties” on page 57

Book General Properties

Property:	Type:	Description:
BookDontUpdateReferences	F_Int	False if the FrameMaker product updates cross-references when it opens the book.
BookIsModified	F_Int	True if the book has been modified.
BookIsSelected	F_Int	True if the book icon in the book window is selected.
FirstComponentInBook	FMBookComponent	First FMBookComponent in the FMBook object.
FirstSelectedComponentInBook	FMBookComponent	First selected FMBookComponent in the book.
IsIconified	F_Int	True if the book window is iconified.
IsInFront	F_Int	True if the book window is in front of other windows in the FrameMaker product session.
IsOnScreen	F_Int	True if the book is visible on the screen. Note that this property is always True for books, and setting it to False has no effect.

Property:	Type:	Description:
Label	F_String	The title in the book window title bar.
Name	F_String	Pathname of the book.
NextOpenBookInSession	FMBook	ID of the next open FMBook in session's list of open books.
ScreenHeight	F_Int	Height of the book window in pixels.
ScreenWidth	F_Int	Width of the book window in pixels.
ScreenX	F_Int	The offset of the book window in pixels from the left side of the screen (or the left of the FrameMaker product application window on Windows). If you set a value that would result in the book window being off the screen, that value is ignored and the old value is retained.
ScreenY	F_Int	The offset of the book window in pixels from the top of the screen (or the top of the FrameMaker product application window on Windows). If you set a value that would result in the book window being off the screen, that value is ignored and the old value is retained.
StatusLine	F_String	String that appears in the book status bar. In versions before 6.0, this property always returned an empty string; In versions 6.0 and later, it returns the current status string. If you set string content to StatusLine (a string other than an empty string), the string will remain in the status bar until you reset it. To reset StatusLine so the FrameMaker product automatically updates the status line with normal status information, set it to an empty string ("").

Property:	Type:	Description:
TypeOfDisplayText	F_Int	The type of text snippet to display for each icon in the book window: mFV_BK_FILENAME displays the book component's filename mFV_BK_TEXT displays the first paragraph of the component's first flow
BookComponents	Array of FMBookComponent	The list of FMBookComponent objects that are managed by the current FMBook object.

Book PDF Properties

FMBook objects have a number of properties to provide PDF information.

The FP_PDFDocInfo property defines a list of strings to enter in a PDF Document Info dictionary. For one dictionary entry you provide two strings; the first is the entry name, and the second is the entry content. The entry name can be up to 126 bytes; the entry content can be up to 32765 characters.

The entry name is a string of bytes within the ASCII range. For non-printable ASCII, provide Hex codes. To represent a Hex code, precede the code with the "#" character; for example #24. To represent the "#" character, enter #23. Finally, an entry name may not include a byte with a value of zero (#00).

The entry content can include Unicode encoding. However, FrameMaker doesn't support support Unicode in these strings. To enter Unicode characters in these strings, you must use an ASCII representation of Unicode that the FrameMaker product understands, as the following table shows:

To represent these characters:	Use:
Unicode characters within the standard ASCII range, with the exception of the ampersand (&) character	Standard ASCII characters
Unicode characters outside the standard ASCII range	A token (&#x) to identify a Unicode character followed by the hexadecimal value of the Unicode character; for example, &#xC2A7
Ampersand	&#x0026

For more information on Acrobat specific functionality see the Acrobat Documentation.

The following table lists PDF properties for books.

Property:	Type:	Description:
AcrobatBookmarkDisplayTags	F_Int	True if Include Paragraph Tags in Bookmark Text is on (the paragraph tag is added before the paragraph text in each bookmark).
DocAcrobatColumnArticleThreads	F_Int	True if you want separate article threads for each column, False if you want separate article threads for each text frame. Note that DocPDFNoArticleThread must be false.
DocAcrobatDefaultsChanged	F_Int	True if the default heuristics for determining the paragraph level are disabled.
DocAcrobatElementList	Array of F_String	List of the element tags and context labels to include in bookmarks. DocPDFElementList applies only to structured FrameMaker documents.
DocAcrobatElements	F_Int	True if elements rather than paragraphs are used for bookmarks. DocPDFElements applies only to structured FrameMaker documents.
DocAcrobatNoArticleThreads	F_Int	True if you do not want article threads in the resulting PDF.

Property:	Type:	Description:
GenerateAcrobatInfo	F_Int	True if Generate Adobe Acrobat Data is on. To generate PDF data, you must set other document print properties as follows PrintToFile: True PrintThumbnails: False PrintSeps: False PrintBlankPages: True PrintLastSheetFirst: False PrintNumCopies: 1 PrintOddPages: True PrintEvenPages: True PrintScale: 100%
PDFAllNamedDestinations	F_Int	True if PDF generated from this book will include Named Destinations for every paragraph and FrameMaker+SGML structure element in the book. This results in a larger PDF filesize. If False, the generated PDF will have Named Destinations only for those paragraphs and structure elements that have already been marked with PDFDestsMarked = True. For documents created in versions earlier than 6.0, True indicates the user has not used the Optimize PDF Size command for the document; normally your client should not change this value to False.
PDFBookmark	F_Int	True if the FrameMaker product will generate bookmarks when saving as PDF.

Property:	Type:	Description:
PDFBookmarksOpenLevel	F_Int	The level of bookmarks to have expanded when Acrobat opens the generated PDF document. Can be any integer, or one of the following defined values: mFV_PDFBookmarksOpenDefaultLevel mFV_PDFBookmarksOpenAllLevels mFV_PDFBookmarksOpenNoneLevel If you specify an integer is greater than the number of levels in the Bookmarks Settings, mFV_PDFBookmarksOpenAllLevels takes effect.
PDFConvertCMYKtoRGB	F_Int	When True, corresponds with setting Convert CMYK colors to RGB in the Save As PDF dialog box. This property can be True for a document on any platform, but it only has an effect for PDF generated on the Macintosh.
PDFDestsMarked	F_Int	True if the document has paragraphs or elements marked via MarkedForNamedDestination. One of two things must happen in order for this property to be True: The document was created in version 6.0 or later; the document was opened in version 6.0 or later, and the PDF FileSize Optimization client was run over it to mark all paragraphs or elements that are targets of hypertext links. Normally, your client should not set this value.

Property:	Type:	Description:
PDFDistillerAbsent	F_Int	A value of 1 indicates that Acrobat Distiller is not available for the current session.
PDFDocInfo	Array of F_String	A list of strings expressing values to be set in the PDF Document Info dictionary when you save the book as PDF. Each dictionary entry is expressed as a pair of strings; the first string expresses the field name, and the second string expresses the field value. See "XMP Data in FrameMaker Documents" on page 419 of Appendix B, "FrameMaker Document and Session Architecture."
PDFEndPage	F_String	The last page of the printing page range, in the FrameMaker numbering style.
PDFJobOption	F_String	The name of the Distiller Job Options. If the specified name does not exist in the Distiller Job Options list, then the first Distiller Job Option in the list is used.
PDFJobOptionsAbsent	F_Int	A value of 1 indicates that PDF Job Options are not available.
PDFOpenPage	F_String	The PDF page number, in the FrameMaker numbering style, at which Acrobat opens the generated PDF document.
PDFPageHeight	F_Metric	Page height for the generated PDF.
PDFPageWidth	F_Metric	Page width for the generated PDF.
PDFPrintPageRange	F_Int	True for generating PDF for the the specified page range; if False, FrameMaker generates PDF for the entire document or book.

Property:	Type:	Description:
PDFRegistrationMarks	F_Int	Registration marks for the generated PDF. May be one of: mFV_PDFRegistrationMarksNone mFV_PDFRegistrationMarksWestern mFV_PDFRegistrationMarksTombo
PDFSeparateFiles	F_Int	True, if a separate PDF file should be generated for each document in a book. This property can be set for single documents, but is ignored in that case.
PDFStartPage	F_String	The first page of the printing page range, in the FrameMaker numbering style
PDFZoomFactor	F_Metric	When PDFZoomType is mFV_PDFZoomNone, the zoom percentage of the PDF document (metric 25% to 1600%. If the number is negative or zero, mFV_PDFZoomDefault takes effect.
PDFZoomType	F_Int	The PDF zoom setting with which Acrobat opens the generated PDF document. Can be one of: mFV_PDFZoomDefault mFV_PDFZoomPage mFV_PDFZoomWidth mFV_PDFZoomHeight mFV_PDFZoomNone If a different value is specified, mFV_PDFZoomDefault takes effect.

Book Print Properties

Property:	Type:	Description:
PrintBlankPages	F_Int	True if PageRounding allows empty pages at the end of documents.

Property:	Type:	Description:
PrintCollated	F_Int	True if Collate is enabled.
PrintEmulsion	F_Int	Direction of print emulsion: mFV_EMUL_UP: Emulsion side up mFV_EMUL_DOWN: Emulsion side down
PrinterName	F_String	Name of the current printer.
PrintEvenPages	F_Int	True if Print Even-Numbered Pages is enabled.
PrintFileName	F_String	Filename of file to print to. When you set PrintFileName, you can set the filename to the default filename by specifying NULL.
PrintImaging	F_Int	Type of print imaging: mFV_IMG_POSITIVE mFV_IMG_NEGATIVE
PrintLastSheetFirst	F_Int	True if Last Sheet First is enabled.
PrintLowRes	F_Int	True if Low-Resolution is enabled.
PrintNumCopies	F_Int	Number of copies to print.
PrintOddPages	F_Int	True if Odd-Numbered Pages is enabled.
PrintPaperHeight	F_Metric	Height of paper.
PrintPaperWidth	F_Metric	Width of paper.
PrintRegistrationMarks	F_Int	True if Registration Marks is enabled.
PrintScale	F_Metric	Scale factor expressed as a percentage of black (metric 0% to 100%)
PrintSeps	F_Int	True if Print Separations is enabled.
PrintToFile	F_Int	True if Print Only to File is enabled.
SkipBlankSeps	F_Int	True if Skip Blank Separations is enabled (don’t print blank color separations).

Book Structure Properties

Property:	Type:	Description:
CustomElementList	Array of F_String	List of tags to display when ElementCatalogDisplay is set to mFV_ELCAT_CUSTOM.
ElementCatalog	Array of udtElementCatalogEntry	List of elements in Element Catalog.
ElementCatalogDisplay	F_Int	Catalog display options. Show tags for: mFV_ELCAT_STRICT: valid children for working start to finish mFV_ELCAT_LOOSE: valid children for working in any order mFV_ELCAT_CHILDREN: children allowed anywhere in parent mFV_ELCAT_ALL: all elements mFV_ELCAT_CUSTOM: the list of tags specified by CustomElementList
ElementSelection	udtElementRange	The currently selected element range in the book.
FileExtensionOverride	F_String	The file extension to use when saving this document as XML. Typically, this is used to save XHTML with a .htm extension rather than .xml. This setting should be set in the structure application for this document's DOCTYPE, and that setting may be a more reliable source for this value.
FirstElementDefInDoc	FMElement	First FMElementDefinition object in book.
FirstFmtChangeListInDoc	FMFormatChangeList	First FMFormatChangeList in the list of format change lists in the book.
HighestLevelElement	FMElement	Highest-level FMElement, if the book is structured.

Property:	Type:	Description:
NewElemAttrDisplay	F_Int	Specifies attribute display properties for new elements: mFV_ATTR_DISP_NONE: don't display attributes mFV_ATTR_DISP_REQSPEC: display required and specified attributes mFV_ATTR_DISP_ALL: display all attributes
NewElemAttrEditing	F_Int	Specifies when the Edit Attributes dialog box appears for new elements: mFV_ATTR_EDIT_NONE mFV_ATTR_EDIT_REQUIRED mFV_ATTR_EDIT_ALWAYS
SeparateInclusions	F_Int	True if inclusions are listed separately in the element catalog.
StructuredApplication	F_String	The name of the structure application that is associated with the book. If the book has no associated structure application, this property contains an empty string.
UseInitialStructure	F_Int	True if FrameMaker+SGML inserts initial structure for new elements.
XmlDocType	F_String	The DOCTYPE parameter from the source XML.

Property:	Type:	Description:
XmlEncoding	F_String	The encoding parameter of the XML Declaration for the source XML. The string is empty if no encoding is specified. If this property is set, the XML Declaration will contain the encoding parameter with this value on Save As XML. For information about the encoding that FrameMaker+SGML supports with a default installation, see the online manual, the Structure Application Developer's Guide.
XmlFileEncoding	F_String	The encoding that was detected for the source XML book. If no encoding was specified for the source XML, XmlEncoding will be an empty string. In that case, if this string is set it will determine the encoding to use when saving as XML. If XmlEncoding has a value, this string may be empty. For information about the encoding that FrameMaker+SGML supports with a default installation, see the online manual, the Structure Application Developer's Guide.
XmlPublicId	F_String	The DOCTYPE public identifier for the source XML document.

Property:	Type:	Description:
XmlStandAlone	F_Int	An integer that specifies the XML standalone parameter for the XML document that was the source of this document. Can be one of mFV_XML_STANDALONE_YES mFV_XML_STANDALONE_NO mFV_XML_STANDALONE_NA The standalone parameter is declared in the XML Declaration. For a file with no XML Declaration, the value is mFV_XML_STANDALONE_NODEC. For an XML Declaration with no standalone parameter, this value is mFV_XML_STANDALONE_NONE.
XmlStyleSheet	F_String	The XML stylesheet processing instruction to write out to XML when saving the book as XML. The FDK does not verify that you use correct syntax in this string. The string you set should not include the PI delimiters, <? and ?>. For example, the string you supply for my.css would be: `"type=\"text\\css\" href=\"my.css\""` Only use this string to set a specific stylesheet specification. Getting this parameter always returns NULL. To get the list of stylesheet specifications associated with a book, use XmlStyleSheetList, below.

Property:	Type:	Description:
XmlStyleSheetList	Array of F_String	A list of stylesheet processing instructions for the current book. One book can have more than one stylesheet specification associated with it. The strings should not include the PI delimiters, <? and ?>. For example, the string you supply for my.css would be: `"type=\"text\\css\" href=\"my.css\""` Setting a list to XmlStyleSheetList completely overwrites the preceeding list.
XmlSystemId	F_String	The DOCTYPE system identifier for the source XML document.
XmlUseBOM	F_Int	Indicates whether a byte order mark was detected when opening the source XML. Can be one of: mFV_XML_USEBOM_NA mFV_XML_USEBOM_NO mFV_XML_USEBOM_YES When saving as XML, it this is set to mFV_XML_USEBOM_YES, FrameMaker writes a byte order mark in the resulting XML.
XmlVersion	F_String	The XML Version that was specified in the XML Declaration when the file was opened. If no XML version was specified, this property is set to an empty string. If this string contains an invalid XML declaration, it will produce a parsing error when this book is saved as XML.

Property:	Type:	Description:
XmlWellFormed	F_Int	Indicates whether the source XML qualified as well formed. Can be one of: mFV_XML_WELLFORMED_NA mFV_XML_WELLFORMED_NO mFV_XML_WELLFORMED_YES

Book View Only Properties

Property:	Type:	Description:
BookIsViewOnly	F_Int	True if the book is view-only.
ViewOnlyDeadCodes	Array of F_Int (unsigned)	List of FrameMaker F-codes that can't be executed in the book when it's View Only.
ViewOnlyWinBorders	F_Int	True if the book has normal window borders; False if the book's border buttons are suppressed
ViewOnlyWinPopup	F_Int	True if the book window pop-up menu is available

Methods:

Method:	Description:
Compare	Compares two book files.
NewBookComponentInHierarchy	Adds a new FMBookComponent to the book at a specified location.
NewBookComponent	Adds a new FMBookComponent to the book.
Save	Saves the book file using a property list to script the save.
SimpleSave	Saves the book file.
Print	Prints the book
Generate	Generates the content for generated files.
ImportElementDefs	Imports elements definitions from the specified file into the book file.

Methods: (Continued)

Method:	Description:
ImportFormats	Imports formats from the specified document into the specified book components.
UpdateBook	Updates the book using a property list to script the update.
Close	Closes the book.

FMBookComponent

Details

A book component is the item in the FMBook object that corresponds to a specific FMDocument. You can use FMBookComponent objects to access the actual document files.

See also

"FMObject" on page 166, "FMBook" on page 42, and "FMDocument" on page 88

Properties

Property:	Type:	Description:
BookComponentIsGeneratable	F_Int	True if book component is a generated file (BookComponentType is not set to mFV_BK_NOT_GENERATABLE)
BookComponentType	F_Int	Type of book component: mFV_BK_INDEX_AUTHOR: index of authors mFV_BK_INDEX_FORMATS: index of formats mFV_BK_INDEX_MARKER: index of markers mFV_BK_INDEX_REFERENCES: index of references mFV_BK_INDEX_STAN: standard index mFV_BK_INDEX_SUBJECT: subject index mFV_BK_LIST_FIGURE: list of figures mFV_BK_LIST_FORMATS: list of formats mFV_BK_LIST_MARKER: list of markers mFV_BK_LIST_MARKER_ALPHA: alphabetical list of markers mFV_BK_LIST_PGF: list of paragraphs mFV_BK_LIST_PGF_ALPHA: alphabetical list of paragraphs mFV_BK_LIST_REFERENCES: list of references mFV_BK_LIST_TABLE: list of tables mFV_BK_NOT_GENERATABLE: book component is not a generated file mFV_BK_TOC: table of contents
BookParent	FMBook	FMBook that contains the component.

Property:	Type:	Description:
ChapNumComputeMethod	F_Int	The component document's chapter numbering type: mFV_NUM_CONTINUE: Continue the numbering from the previous chapter. mFV_NUM_RESTART: Use the value specified for FP_ChapterNumber. mFV_NUM_SAME: Use the same chapter number as for the previous file. mFV_NUM_READ_FROM_FILE: Use the numbering properties from the document associated with this book component.
ChapterNumber	F_Int	If ChapNumComputeMethod is mFV_NUM_RESTART, use this as the chapter number
ChapterNumStyle	F_Int	The numbering style; one of: mFV_NUMSTYLE_NUMERIC: Arabic. mFV_NUMSTYLE_ROMAN_UC: Roman, uppercase. mFV_NUMSTYLE_ROMAN_LC: Roman,lowercase. mFV_NUMSTYLE_ALPHA_UC: Alphabetic, uppercase. mFV_NUMSTYLE_ALPHA_LC: Alphabetic, lowercase. mFV_NUMSTYLE_KANJI: Kanji. mFV_NUMSTYLE_ZENKAKU: Zenkaku. mFV_NUMSTYLE_ZENKAKU_UC: Zenkaku, uppercase. mFV_NUMSTYLE_ZENKAKU_LC: Zenkaku, lowercase. mFV_NUMSTYLE_KANJI_KAZU: Kazu. mFV_NUMSTYLE_DAIJI: Daiji. mFV_NUMSTYLE_TEXT: Text.

Property:	Type:	Description:
ChapterNumText	F_String	If ChapNumStyle is mFV_NUMSTYLE_TEXT, use this string as the chapter number.
ComponentDisplayText	F_String	Text that displays in the book window when the value of TypeOfDisplayText = mFV_BK_TEXT. To reset ComponentDisplayText so the FrameMaker product automatically updates the text line with normal information, set it to an empty string ("").
ComponentElement	FMElement	The FMElement that corresponds to this book component (structured FrameMaker, only)
ComponentIsSelected	F_Int	True if the component is selected in the book window
ExtractElementTags	Array of F_String	List of element tags that are used to set up a generatable file (table of contents, list of figures, or list of tables)
ExtractTags	Array of F_String	List of paragraph tags or markers type names that are used to set up a generatable file (table of contents, list of figures, standard index or index of authors)
FirstPageNum	F_Int	Number for the first page in the component; used when PageNumComputeMethod = mFV_NUM_RESTART.
FnCustNumString	F_String	Characters for custom document footnote numbers.
FnFirstNum	F_String	Number for the first footnote in the component; used when FnNumComputeMethod = mFV_NUM_RESTART.

Property:	Type:	Description:
FnNumComputeMethod	F_Int	The component document's footnote numbering type: mFV_NUM_CONTINUE: Continue the numbering from the previous file. mFV_NUM_RESTART: Use the number specified by FP_FnFirstNum. mFV_NUM_PER_PAGE: Restart numbering on each page. mFV_NUM_READ_FROM_FILE: Use the numbering properties from the document associated with this book component.
FnNumStyle	F_Int	Footnote numbering style: mFV_FN_NUM_NUMERIC: Arabic mFV_FN_NUM_ROMAN_UC: Roman uppercase mFV_FN_NUM_ROMAN_LC: Roman lowercase mFV_FN_NUM_ALPHA_UC: Alphabetic uppercase mFV_FN_NUM_ALPHA_LC: Alphabetic lowercase mFV_FN_NUM_KANJI: Kanji characters mFV_FN_NUM_ZENKAKU: Zenkaku mFV_FN_NUM_ZENKAKU_UC: Zenkaku uppercase mFV_FN_NUM_ZENKAKU_LC: Zenkaku lowercase mFV_FN_NUM_KANJI_KAZU: Kazu FmV_FN_NUM_DAIJI: Daiji mFV_FN_NUM_CUSTOM: Custom numbering
GenerateInclude	F_Int	True if the document appears in the scroll list of files to be generated by the Generate/Update command for the book

Property:	Type:	Description:
ImportFmtInclude	F_Int	True if the book component is included in the list of components to be updated with imported formats or element definitions when the user or a client executes Import Formats or Import Element Definitions.
InsertLinks	F_Int	True if this component is generatable, and hypertext links are automatically inserted.
Name	F_String	Pathname of the FMDocument that the component represents.
NextComponentInBook	FMBookComponent	Next FMBookComponent in the book file.
NextSelectedComponentInBook	FMBookComponent	Next selected FMBookComponent in the book window.
PageNumComputeMethod	F_Int	The component document's page numbering type: mFV_NUM_CONTINUE: Continue the numbering from the previous file. mFV_NUM_RESTART: Restart numbering at the value specified by FirstPageNum. mFV_NUM_READ_FROM_FILE: Use the numbering properties from the document associated with this book component.

Property:	Type:	Description:
PageNumStyle	F_Int	Page numbering style: mFV_PAGE_NUM_NUMERIC: Arabic mFV_PAGE_NUM_ROMAN_UC: Roman uppercase mFV_PAGE_NUM_ROMAN_LC: Roman lowercase mFV_PAGE_NUM_ALPHA_UC: Alphabetic uppercase mFV_PAGE_NUM_ALPHA_LC: Alphabetic lowercase mFV_PAGE_NUM_KANJI: Kanji characters mFV_PAGE_NUM_ZENKAKU: Zenkaku mFV_PAGE_NUM_ZENKAKU_UC: Zenkaku uppercase mFV_PAGE_NUM_ZENKAKU_LC: Zenkaku lowercase mFV_PAGE_NUM_KANJI_KAZU: Kazu mFV_PAGE_NUM_DAIJI: Daiji
PagePrefix	F_String	Obsolete for version 6.0 and later; use chapter and volume numbering instead: Page prefix string
PageSide	F_Int	Page side to start the component document on: mFV_BK_START_FROM_FILE mFV_BK_START_NEXT_AVAILABLE mFV_BK_START_LEFT mFV_BK_START_RIGHT
PageSuffix	F_String	Obsolete for version 6.0 and later; use chapter and volume numbering instead: Page suffix string

Property:	Type:	Description:
PgfNumComputeMethod	F_Int	The component document's paragraph numbering type: mFV_NUM_CONTINUE: Continue the numbering from the previous file. mFV_NUM_RESTART: Restart numbering at 1. mFV_NUM_READ_FROM_FILE: Use the numbering properties from the document associated with this book component.
PrevComponentInBook	FMBookComponent	Previous FMBookComponent in the book file
PrintInclude	F_Int	True if the component document is included in list of book files to be printed
TblFnCustNumString	F_String	Characters for custom table footnote numbers.
TblFnNumComputeMethod	F_Int	The component document's table footnote numbering type: mFV_NUM_RESTART: Start at 1. mFV_NUM_READ_FROM_FILE: Use the numbering properties from the document associated with this book component.

Property:	Type:	Description:
TblFnNumStyle	F_Int	Table footnote numbering style: mFV_FN_NUM_NUMERIC: Arabic mFV_FN_NUM_ROMAN_UC: Roman uppercase mFV_FN_NUM_ROMAN_LC: Roman lowercase mFV_FN_NUM_ALPHA_UC: Alphabetic uppercase mFV_FN_NUM_ALPHA_LC: Alphabetic lowercase mFV_FN_NUM_KANJI: Kanji characters mFV_FN_NUM_ZENKAKU: Zenkaku mFV_FN_NUM_ZENKAKU_UC: Zenkaku uppercase mFV_FN_NUM_ZENKAKU_LC: Zenkaku lowercase mFV_FN_NUM_KANJI_KAZU: Kazu mFV_FN_NUM_DAIJI: Daiji mFV_FN_NUM_CUSTOM: Custom numbering
Unique	F_Int	UID of the book component
VolNumComputeMethod	F_Int	The component document's volume numbering type: mFV_NUM_CONTINUE: Continue the numbering from the previous volume. mFV_NUM_RESTART: Use the value specified for FP_VolumeNumber. mFV_NUM_SAME: Use the same volume number as for the previous file. mFV_NUM_READ_FROM_FILE: Use the numbering properties from the document associated with this book component.
VolumeNumber	F_Int	If VolNumComputeMethod is mFV_NUM_RESTART, use this as the volume number

Property:	Type:	Description:
VolumeNumStyle	F_Int	The numbering style; one of: mFV_NUMSTYLE_NUMERIC: Arabic. mFV_NUMSTYLE_ROMAN_UC: Roman, uppercase. mFV_NUMSTYLE_ROMAN_LC: Roman, lowercase. mFV_NUMSTYLE_ALPHA_UC: Alphabetic, uppercase. mFV_NUMSTYLE_ALPHA_LC: Alphabetic, lowercase. mFV_NUMSTYLE_KANJI: Kanji. mFV_NUMSTYLE_ZENKAKU: Zenkaku. mFV_NUMSTYLE_ZENKAKU_UC: Zenkaku, uppercase. mFV_NUMSTYLE_ZENKAKU_LC: Zenkaku, lowercase. mFV_NUMSTYLE_KANJI_KAZU: Kazu. mFV_NUMSTYLE_DAIJI: Daiji. mFV_NUMSTYLE_TEXT: Text.
VolumeNumText	F_String	If VolNumStyle is mFV_NUMSTYLE_TEXT, use this string as the chapter number.

FMCell

Details

FMCell objects represent the cells in a table or table row.

See also

"FMObject" on page 166, "FMTable" on page 205, and "FMRow" on page 191.

Properties

Property:	Type:	Description:
CellAboveInCol	FMCell	FMCell above current cell.
CellAngle	F_Int	Angle of rotation.
CellBelowInCol	FMCell	FMCell below current cell.
CellColNum	F_Int	Cell's column number.
CellDefaultBottomRuling	FMRulingFormat	FMRulingFormat for the cell's default bottom ruling.
CellDefaultLeftRuling	FMRulingFormat	FMRulingFormat for the cell's default left ruling).
CellDefaultRightRuling	FMRulingFormat	FMRulingFormat for the cell's default right ruling.
CellDefaultTopRuling	FMRulingFormat	FMRulingFormat for the cell's default top ruling.
CellIsShown	F_Int	True if the cell is conditional and is visible.
CellIsStraddled	F_Int	If the cell is in a straddle but is not the first cell, CellIsStraddled is True. If the cell is the first cell in a straddle, or it is not in a straddle, CellIsStraddled is False.
CellNumColsStraddled	F_Int	If the cell is the first cell in a horizontal straddle, CellNumColsStraddled is the number of columns in the straddle. Otherwise, it is 1.
CellNumRowsStraddled	F_Int	If the cell is the first cell in a vertical straddle, CellNumRowsStraddled is the number of rows in the straddle. Otherwise, it is 1.
CellOverrideBottomRuling	FMRulingFormat	FMRulingFormat for the cell's bottom ruling. NULL if there is no override.
CellOverrideFill	F_Int	Fill pattern. NULL if there is no override fill pattern.
CellOverrideLeftRuling	FMRulingFormat	FMRulingFormat for the cell's left ruling. NULL if there is no override left ruling.
CellOverrideRightRuling	FMRulingFormat	FMRulingFormat for the cell's right ruling. NULL if there is no override right ruling.

Property:	Type:	Description:
CellOverrideShading	FMColor	FMColor for the override shading of the cell. NULL if there is no override shading.
CellOverrideTopRuling	FMRulingFormat	FMRulingFormat for the cell's top ruling. NULL if there is no override top ruling.
CellRow	FMRow	FMRow containing the cell.
CellUseOverrideBRuling	F_Int	True if the cell's bottom ruling (specified by CellOverrideBottomRuling) overrides the default ruling specified by the table format.
CellUseOverrideFill	F_Int	True if the cell's fill pattern (specified by CellOverrideFill) overrides the default fill pattern specified by the table format.
CellUseOverrideLRuling	F_Int	True if the cell's left ruling (specified by CellOverrideLeftRuling) overrides the ruling specified by the table format.
CellUseOverrideRRuling	F_Int	True if the cell's right ruling (specified by CellOverrideRightRuling) overrides the ruling specified by the table format.
CellUseOverrideShading	F_Int	True if the cell's shading (specified by CellOverrideShading) overrides the default shading specified by the table format.
CellUseOverrideTRuling	F_Int	True if the cell's top ruling (specified by CellOverrideTopRuling) overrides the default top ruling specified by the table format.
ContentHeight	F_Metric	The distance between the top of the cell and the baseline of the last line in the cell.
Element	FMElement	If the cell is in a FrameMaker+SGML document, the FMElement containing the cell.
FirstPgf	FMParagraph	First FMParagraph paragraph in the cell.
HighestLevelElement	FMElement	FMElement for the cell's highest-level element if the cell is in a structured FrameMaker+SGML document. HighestLevelElement is obsolete but is supported for backward compatibility.

Property:	Type:	Description:
InTextFrame	FMTextFrame	FMTextFrame frame containing the cell.
InTextObj	FMSubCol	FMSubCol containing the cell.
LastPgf	FMParagraph	Last FMParagraph in the cell .
NextCell	FMCell	Next FMCell in the text frame.
NextCellInRow	FMCell	Next FMCell in current row from left to right.
NextCellInTbl	FMCell	Next FMCell from left to right. If the cell is at the end of a row, the next cell is the first cell in the next row.
Overflowed	F_Int	Specifies whether the text in the cell overflows. True if the row Height Limit Maximum is too low to display all the text in the cell.
PrevCell	FMCell	Previous FMCell in the text frame.
PrevCellInRow	FMCell	Previous FMCell in current row.
Unique	F_Int	The cell's UID.
Text	F_String	The text content of the cell

Methods:

Method:	Description:
StraddleCells	Straddles the specified cells.
UnStraddleCells	Unstraddles the specified straddled cells
DeleteCellText	Clears the text from a cell
IsCellEmpty	Returns True if the specified cell has no content

FMCharacterFormat

Details

A character format is a set of character formatting properties. The character format is stored in the Character Format Catalog, and can be applied to a range of text.

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
Capitalization	F_Int	The capitalization type: mFV_CAPITAL_CASE_NORM: normal capitalization (mixed uppercase and lowercase) mFV_CAPITAL_CASE_SMALL: small caps mFV_CAPITAL_CASE_LOWER: lowercase letters only mFV_CAPITAL_CASE_UPPER: uppercase letters only
ChangeBar	F_Int	True if Change Bars are on.
CharTag	F_String	The character format's tag name.
Color	FMColor	Spot color (FMColor).
CombinedFontFamily	FMCombinedFont Definition	The FMCombinedFontDefinition for the character format.
FontAngle	F_Int	Font angle (specifies an index into the array of font angles provided by the session property FontAngleNames).
FontEncodingName	F_String	The font's encoding.
FontFamily	F_Int	Font family (specifies an index into the array of font families provided by the session property FontFamilyNames).
FontPlatformName	F_String	Name that uniquely identifies a font on a specific platform. For combined fonts, this is the Asian font name.
FontPostScriptName	F_String	Name given to a font when it is sent to a PostScript printer. For combined fonts, this is the Asian font name.
FontSize	F_Metric	Font size (2 pt to 400 pt).
FontVariation	F_Int	Font variation (specifies an index into the array of font variations provided by the session property FontVariationNames).

Property:	Type:	Description:
FontWeight	F_Int	Font weight (specifies an index into the array of font weights provided by the session property FontWeightNames).
KernX	F_Metric	Horizontal kern value for manual kerning expressed as a percentage of an em (metric –1000% to 1000%). A positive value moves a character right and a negative value moves a character left.
KernY	F_Metric	Vertical kern value for manual kerning expressed as a percentage of an em (metric –1000% to 1000%). A positive value moves characters up and a negative value moves characters down.

Property:	Type:	Description:
Language	F_Int	Hyphenation and spell-checking language to use: mFV_LANG_BRAZILIAN mFV_LANG_BRITISH mFV_LANG_CANADIAN_FRENCH mFV_LANG_CATALAN mFV_LANG_DANISH mFV_LANG_DUTCH mFV_LANG_ENGLISH mFV_LANG_FINNISH mFV_LANG_FRENCH mFV_LANG_GERMAN mFV_LANG_ITALIAN mFV_LANG_NOLANGUAGE mFV_LANG_NORWEGIAN mFV_LANG_NYNORSK mFV_LANG_PORTUGUESE mFV_LANG_SPANISH mFV_LANG_SWEDISH mFV_LANG_SWISS_GERMAN mFV_LANG_JAPANESE mFV_LANG_TRADITIONAL_CHINESE mFV_LANG_SIMPLIFIED_CHINESE mFV_LANG_KOREAN
Name	F_String	The character format's name.
NextCharFmtInDoc	FMCharacterFormat	Next FMCharacterFormat format in document.
Overline	F_Int	True if Overline is enabled.
PairKern	F_Int	True if Pair Kern is enabled.

Property:	Type:	Description:
Position	F_Int	Vertical position of character: mFV_POS_NORM: Normal mFV_POS_SUB: Subscript mFV_POS_SUPER: Superscript
Spread	F_Metric	Obsolete property, but still functional. See corresponding "tracking" property below.
Stretch	F_Metric	Character stretch (set width) expressed as a percentage of normal stretch for the font (metric –10% to 1000%).
Strikethrough	F_Int	True if Strikethrough is enabled.
Underlining	F_Int	Underlining type: mFV_CB_NO_UNDERLINE mFV_CB_SINGLE_UNDERLINE mFV_CB_DOUBLE_UNDERLINE mFV_CB_NUMERIC_UNDERLINE
UseCapitalization	F_Int	True if the Capitalization property overrides default; False if As Is setting used.
UseChangeBar	F_Int	True if the ChangeBar property overrides default; False if As Is setting used.
UseColor	F_Int	True if the Color property overrides default; False if As Is setting used.
UseFontAngle	F_Int	True if the FontAngle overrides default; False if As Is setting used.
UseFontFamily	F_Int	True if the FontFamily overrides default; False if As Is setting used.
UseFontSize	F_Int	True if the FontSize overrides default; False if As Is setting used.
UseFontVariation	F_Int	True if the FontVariation overrides default; False if As Is setting used.
UseFontWeight	F_Int	True if the FontWeight overrides default; False if As Is setting used.
UseKernX	F_Int	True if the KernX overrides default; False if As Is setting used.

Property:	Type:	Description:
UseKernY	F_Int	True if the KernY overrides default; False if As Is setting used.
UseOverline	F_Int	True if the Overline property overrides default; False if As Is setting used.
UsePairKern	F_Int	True if the PairKern property overrides default; False if As Is setting used.
UsePosition	F_Int	True if the Position overrides default; False if As Is setting used.
UseSpread	F_Int	Obsolete property, but still functional. See corresponding "tracking" property below.
UseStretch	F_Int	True if the Stretch property overrides default, False if As Is setting is used.
UseStrikethrough	F_Int	True if the Strikethrough property overrides default; False if As Is setting used.
UseUnderlining	F_Int	True if the Underlining property overrides default; False if As Is setting used.
WesternFontPlatformName	F_String	Name that uniquely identifies the Roman component of a combined font on a specific platform.
WesternFontPostScriptName	F_String	Name given to the Roman component of a combined font when it is sent to a PostScript printer.

FMColor

Details

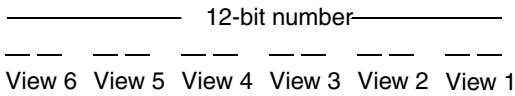
FMColor objects represent each color in a FrameMaker document.

The ColorViewCtl property specifies a 12-bit number for color views. The two least significant bits are View 1, the next 2 bits are View 2, and so on, as shown below.

The values of each 2-bit setting are one of the following:

- mFV_SEP_NORMAL
- mFV_SEP_NONE
- mFV_SEP_WHITE

The following picture illustrates the bit positions representing spot color views:



See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
Black	F_Metric	Percentage of black (metric 0% to 100%)
ColorOverprint	F_Int	Overprint setting for the color: mFV_COLOR_OVERPRINT mFV_COLOR_KNOCKOUT
ColorPrintCtl	F_Int	Type of color printing used in document: mFV_PRINT_SPOT mFV_PRINT_PROCESS mFV_PRINT_NO
ColorTintPercent	F_Metric	The tint percentage (0% to 100%) or mFV_COLOR_NOT_TINTED if the color is not a tint. Specifies the percentage of the TintBaseColor to use for tinting.
ColorViewCtl	F_Int	A 12-bit number for spot color views. The least significant 2 bits are View 1, the next 2 bits are View 2, and so on. For more information, see the details for FMColor.
Cyan	F_Metric	Percentage of cyan (metric 0% to 100%)
FamilyName	F_String	Color library name. Note that that you must specify the full ink name, including any trademark symbols. For example, use `"MUNSELL\xa8 Book of Color"` for "MUNSELL® Book of Color."
InkName	F_String	Specifies the name of the color library pigment. Use this instead of Pantone.
Magenta	F_Metric	Percentage of magenta (metric 0% to 100%)

Property:	Type:	Description:
Name	F_String	Name of color
NextColorInDoc	FMColor	Next FMColor object in document
ReservedColor	F_Int	Color names reserved by FrameMaker: mFV_COLOR_NOT_RESERVED mFV_COLOR_CYAN mFV_COLOR_MAGENTA mFV_COLOR_YELLOW mFV_COLOR_BLACK mFV_COLOR_WHITE mFV_COLOR_RED mFV_COLOR_GREEN mFV_COLOR_BLUE
TintBaseColor	FMColor	FMColor from which the tint is derived, or mFV_NO_BASE_COLOR if the color is not a tint.
Yellow	F_Metric	Percentage of yellow (metric 0% to 100%)

FMCombinedFontDefinition

Details

FMCombinedFontDefinition objects are document objects that define a combination of a Western font for Roman characters, and a non-Western font for Asian characters.

> **Important:** Combined fonts are stored with the document, and not with the session. The FMSession property FontFamilyNames returns a list of the fonts available for the current session, but it does not include any combined fonts. To get the first combined font in a FMDocument, use CombinedFontFamily

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
BaseFamily	F_Int	Asian font family (specifies index into the arrays of font families provided by the FMSession property, FontFamilyNames
FontEncodingName	F_String	Combined font's encoding, based on the BaseFamily
Name	F_String	Name of the combined font.
NextCombinedFontDefnInDoc	F_Int	Next FMCombinedFontDefinition instance in the document
WesternFamily	F_Int	Western font family (specifies index into the arrays of font families provided by the FMSession property, FontFamilyNames)
WesternShift	F_Metric	Baseline offset of Roman text expressed as a percentage of base font size (metric 1% to 1000%)
WesternSize	F_Metric	Scaling factor for Roman text expressed as a percentage of base font size (metric 1% to 1000%)

FMCommand

Details

FMCommand objects represent commands which can appear within FMMenu objects. Commands can be separated in the menu by FMMenuItemSeperator objects.

You can use the EnabledWhen property to specify when to enable or disable the FMCommand. In a menu, a disabled command is greyed out. The following table lists the values for the EnabledWhen that you can use to specify enabling.

EnabledWhen value	Context in which the item is enabled
mFV_ENABLE_ALWAYS_ENABLE	All contexts. This is the default value. If the menu item is disabled, setting EnabledWhen to this value enables it.
mFV_ENABLE_ALWAYS_DISABLE	No context. The menu item is disabled. If a menu item is enabled and you set EnabledWhen to this value, it disables and dims the menu item.

EnabledWhen value	Context in which the item is enabled
mFV_ENABLE_BOOK_HAS_SELECTION	The book contains a selection.
mFV_ENABLE_DOC_OR_BOOK_HAS_SELECTION	A document is in the front, or a book has a selection.
mFV_ENABLE_IN_PARA_TEXT	The insertion point or selection is in a paragraph (but not in a math object).
mFV_ENABLE_IN_TEXT_LINE	The insertion point or selection is in a graphic text line.
mFV_ENABLE_IS_TEXT_SEL	The selection is in a paragraph.
mFV_ENABLE_IN_MATH	The insertion point or selection is in a math object.
mFV_ENABLE_IN_TEXT	The insertion point or selection is in a graphic text line or a paragraph.
mFV_ENABLE_OBJ_PROPS	The insertion point is in text, a table, or a math object, or a graphic object is selected.
mFV_ENABLE_IN_TABLE	The insertion point or selection is in any part of a table.
mFV_ENABLE_IN_TABLE_TITLE	The insertion point or selection is in the table title.
mFV_ENABLE_IN_CELL_TEXT	The insertion point or selection is in a table cell.
mFV_ENABLE_IS_CELL	A single cell in a table is selected.
mFV_ENABLE_IS_CELLS	One or more cells in a table are selected.
mFV_ENABLE_IS_TABLE	An entire table is selected.
mFV_ENABLE_IS_OBJ	An object is selected.
mFV_ENABLE_IS_TEXT_FRAME	A text frame is selected.
mFV_ENABLE_IS_OR_IN_FRAME	The selected object is a graphic frame or is in a graphic frame that is not a page frame.
mFV_ENABLE_IS_AFRAME	The first selected object is an anchored frame.
mFV_ENABLE_IS_GRAPHIC_INSET	The first selected object is a graphic inset.
mFV_ENABLE_IS_TEXT_INSET	The first selected object is a text inset.
mFV_ENABLE_IN_FLOW	A text frame is selected, or the insertion point or selection is in a paragraph.
mFV_ENABLE_COPY	Some text or an object is selected.
mFV_ENABLE_COPY_FONT	The insertion point or selection is in the text of a paragraph, a math object, a table, or a text line.

EnabledWhen value	Context in which the item is enabled
mFV_ENABLE_CAN_PASTE	The Clipboard contains an object or text that can be pasted at the insertion point.
mFV_ENABLE_IS_VIEW_ONLY	The current document is locked.
mFV_ENABLE_NEEDS_DOCP_ OR_BOOKP	A document or a book is open.
mFV_ENABLE_NEEDS_DOCP_ONLY	A document is open.
mFV_ENABLE_NEEDS_BOOKP_ONLY	A book is open.

See also

"FMObject" on page 166, "FMCommandObject" on page 84.

Properties

Property:	Type:	Description:
CanHaveCheckMark	F_Int	True if the menu item can have a check mark. If the menu item is defined by the FrameMaker product, you can get this property, but not set it.
CheckMarkIsOn	F_Int	True if the menu item can have a check mark and the check mark is on. If the menu item is defined by the FrameMaker product, you can get this property, but not set it.
CommandNum	F_Int	The integer that you specified for the Id argument of the DefineCommand method when you created the command. When the user executes the command, the FrameMaker product passes this integer to your client's function. See the OnApiCommand event for more information. If the menu item is defined by the FrameMaker product, you can get this property, but not set it.

Property:	Type:	Description:
EnabledWhen	F_Int	The context in which the menu item is enabled. For a list of the constants that this field can specify, see the table above. If the menu item is defined by the FrameMaker product, you can get this property, but not set it.
ExpandOMaticParent	FMCommand	If the menu item is an expandomatic menu item, the FMCommand that is its virtual parent.
Fcode	F_Int (unsigned)	An f-code that the FrameMaker product executes when the user chooses the menu item or presses the keyboard shortcut.
Fcodes	Array of F_Int (unsigned)	The list of f-codes that the FrameMaker product executes when the user chooses the menu item or presses the keyboard shortcut. Normally, the first f-code in the list is the same as the f-code specified by Fcode.
HasShiftOrUnshiftCommand	F_Int	Specifies whether a command has an accompanying shift command or unshift command: mFV_ITEM_HAS_SHIFT_COMMAND mFV_ITEM_HAS_UNSHIFT_COMMAND mFV_ITEM_HAS_NO_SHIFT_OR_UNSHIFT_COMMAND

Property:	Type:	Description:
HelpLink	F_String	The hypertext link to call when the user requests context-sensitive help for the command. If you set this property, specify the destination file and an optional page number or linkname. For example, specify foo.doc:lastpage. Do not specify hypertext commands such as gotopage. The FrameMaker product automatically prepends the appropriate hypertext command to the HelpLink string when the user requests context-sensitive help. If the destination file is not in the client directory, the FrameMaker product looks for it in the FrameMaker product help directory. This property is valid only for commands created by clients. It is not valid for commands created directly by FrameMaker.
KeyboardShortcutLabel	F_String	The keyboard shortcut string that appears on the menu. This string need not be one of the actual shortcuts specified by KeyboardShortcuts.
KeyboardShortcuts	Array of F_String	The list of keyboard shortcuts the user can press to execute the command. To add a shortcut, append it to the list. You cannot delete shortcuts from the list once the command has been initialized.

Property:	Type:	Description:
Labels	Array of F_String	If the command is a menu item, the list of labels it can have in different contexts. If the menu item has only one label in all contexts, Labels specifies only the string for that label. FrameAC commands can only have a single label in all contexts. If the menu item was created by FrameMaker, and it has different labels in different contexts, Labels specifies pairs of strings with the following format: Context,Label where Label specifies the menu item label and Context specifies the context in which the label appears on the menu. You can modify the Label string, but not the Context string.
MenuItemType	F_Int	The type of command or menu item: mFV_MENUITEM_FRAME: the command is a menu item defined by the FrameMaker product. mFV_MENUITEM_API: the command is a menu item defined by a client. mFV_MENUITEM_EXPANDOMATIC: the menu item is an expandomatic menu item (such as `!ShowParagraphTags`) defined by the FrameMaker product.
NextCommandInSession	FMCommand	The next FMCommand in the list of commands in the session.
ShiftOrUnshiftCommand	FMCommand	If HasShiftOrUnshiftCommand is set to mFV_ITEM_HAS_SHIFT_COMMAND, the ID of the command to use when the user holds down the Shift key. If HasShiftOrUnshiftCommand is set to mFV_ITEM_HAS_UNSHIFT_COMMAND, the ID of the command to use when the user isn't holding down the Shift key.

FMCommandObject

Details

The FMCommandObject is the parent object for FMMenu, FMCommand, or FMMenuItemSeperator objects—these objects inherit properties and methods from FMCommandObject.

FMMenu objects are contained by the FMSession. One menu can contain a number of FMCommand and FMMenuItemSeperator objects.

FMMenuItemSeparator objects use predefined names, which are `!Separator`, `!Separator1`, `!Separator2`, `!Separator3`, `!Separator4`, and `!Separator5`.

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
Label	F_String	The label the user sees on a menu. The label for menu item separators is read-only; it is always ---.
MenuItemIsEnabled	F_Int	True if the menu or menu item is enabled or False if it is disabled (dimmed).
Name	F_String	The command, menu, or menu item separator name.
NextMenuItemInMenu	FMCommandObject	The next menu item, menu, or separator in the menu.
NextMenuItemInSession	FMCommandObject	The next menu item, menu, or separator in the list of menu items, menus, and separators in the session.
PrevMenuItemInMenu	FMCommandObject	The previous menu item, menu, or separator in the menu.

Methods:

Method:	Description:
MenuItemInMenu	Determines whether a menu item or menu is on a menu or menu bar.

Methods: (Continued)

Method:	Description:
AddCommandToMenu	Adds a command to the specified menu.

FMConditionalFormat

Details

FMConditionFormat objects represent the condition formats that are defined for a document. The FMDocument object includes properties to manage conditional text display:

- ShowAll to determine whether all conditions are shown in the document
- ShowCondIndicators to determine whether the contidion indicators are shown in the document

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
CondFmtIsShown	F_Int	True if the condition is shown. To hide text with a specified condition, set this property of the FMConditionalFormat to False, and set the ShowAll property of the FMDocument to False.
Name	F_String	Name of the condition format.
NextCondFmtInDoc	FMConditionalFormat	Next FMConditionalFormat in document.
SepOverride	FMColor	Color separation format override (FMColor).

Property:	Type:	Description:
StyleOverride	F_Int	Style condition indicators for conditional text: mFV_CN_CHANGEBAR mFV_CN_DOUBLE_UNDERLINE mFV_CN_NO_OVERRIDE mFV_CN_OVERLINE mFV_CN_SINGLE_UNDERLINE mFV_CN_STRIKETHROUGH mFV_CN_NUMERIC_UNDERLINE mFV_CN_NMRIC_AND_CHNGBAR
UseSepOverride	F_Int	True if color specified by SepOverride is used instead of default.

FMCrossReference

Details

The FMCrossReference object represents an instance of a cross-reference in the document text.

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
Element	FMElement	If the cross-reference is in a FrameMaker+SGML document, the associated FMElement.
Locked	F_Int	True if the cross-reference is part of a text inset that retains formatting information from the source document. A locked cross-reference is not affected by global formatting performed on the document.
NextXRefInDoc	FMCrossReferenceFormat	Next FMCrossReferenceFormat in document.

Property:	Type:	Description:
TextRange	udtTextRange	Text range that the cross-reference instance encompasses.
Unique	F_Int	The cross-reference's UID.
XRefFile	F_String	The filename of the file containing the cross-reference source. If the cross-reference source is in the same document as the cross reference, the filename is an empty string ("").
XRefFmt	FMCrossReferenceFormat	The cross-reference's format (FMCrossReferenceFormat).
XRefIsUnresolved	F_Int	True if the FrameMaker product was unable to resolve the cross-reference the last time it updated cross-references. Note that this property is set only when the FrameMaker product updates cross-references. Changes to the document, in and of themselves, do not affect this property.
XRefSrcIsElem	F_Int	True if the cross-reference source is a FMElement in a FrameMaker+SGML document.
XRefSrcText	F_String	If XRefSrcIsElem is False, a string specifying UID:pgf_tag:text, where UID is an arbitrary, unique string, pgf_tag is the name of the paragraph format, and text is the text content of the paragraph. If XRefSrcIsElem is True, a string specifying UID:src_name:text, where UID is the value of the ID attribute of the source element, name is the element tag, and text is text content of the source element.

FMCrossReferenceFormat

Details

FMCrossReferenceFormat objects specify the cross-reference formats that are stored in a document..

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
Fmt	F_String	The cross-reference format string (a string that specifies text and building blocks)
Name	F_String	The cross-reference format's name
NextXRefFmtInDoc	FMCrossReferenceFormat	The next FMCrossReferenceFormat in the document.

FMDocument

Details

The FMDocument object represents an open document in a FrameMaker session.

See also

"FMObject" on page 166.

Properties

This reference manual divides the FMDocument properties into the following categories:

- "Acrobat and PDF Properties" on page 89
- "Contained Objects Properties" on page 93
- "Equation Properties" on page 95
- "Footnote Properties" on page 97
- "Hypertext Properties" on page 98
- "General Document Properties" on page 100
- "Changebar Properties" on page 105
- "Condition Properties" on page 105
- "Menu Properties" on page 105
- "Page Properties" on page 105
- "Print Properties" on page 108
- "Rubi Properties" on page 111
- "Selection Properties" on page 111

- "Structure Properties" on page 112
- "Table Footnote Properties" on page 119
- "Type-in Properties" on page 121
- "Typographic Properties" on page 123
- "View Properties" on page 124
- "View-only Properties" on page 126

Acrobat and PDF Properties

Property:	Type:	Description:
AcrobatBookmarkDisplayTags	F_Int	True if Include Paragraph Tags in Bookmark Text is on (the paragraph tag is added before the paragraph text in each bookmark).
DocAcrobatColumnArticleThreads	F_Int	True if you want separate article threads for each column, False if you want separate article threads for each text frame. Note that DocPDFNoArticleThread must be false.
DocAcrobatDefaultsChanged	F_Int	True if the default heuristics for determining the paragraph level are disabled.
DocAcrobatElementList	Array of F_String	List of the element tags and context labels to include in bookmarks. DocPDFElementList applies only to structured FrameMaker+SGML documents.
DocAcrobatElements	F_Int	True if elements rather than paragraphs are used for bookmarks. DocPDFElements applies only to structured FrameMaker+SGML documents.
DocAcrobatNoArticleThreads	F_Int	True if you do not want article threads in the resulting PDF.

Property:	Type:	Description:
GenerateAcrobatInfo	F_Int	True if Generate Adobe Acrobat Data is on. To generate PDF data, you must set other document print properties as follows PrintToFile: True PrintThumbnails: False PrintSeps: False PrintBlankPages: True PrintLastSheetFirst: False PrintNumCopies: 1 PrintOddPages: True PrintEvenPages: True PrintScale: 100%
PDFAllNamedDestinations	F_Int	True if PDF generated from this book will include Named Destinations for every paragraph and FrameMaker+SGML structure element in the document. This results in a larger PDF filesize. If False, the generated PDF will have Named Destinations only for those paragraphs and structure elements that have already been marked with PDFDestsMarked = True.
PDFBookmark	F_Int	True if the FrameMaker product will generate bookmarks when saving as PDF.
PDFBookmarksOpenLevel	F_Int	The level of bookmarks to have expanded when Acrobat opens the generated PDF document. Can be any integer, or one of the following defined values: mFV_PDFBookmarksOpenDefaultLevel mFV_PDFBookmarksOpenAllLevels mFV_PDFBookmarksOpenNoneLevel If you specify an integer is greater than the number of levels in the Bookmarks Settings, mFV_PDFBookmarksOpenAllLevels takes effect.

Property:	Type:	Description:
PDFConvertCMYKtoRGB	F_Int	When True, corresponds with setting Convert CMYK colors to RGB in the Save As PDF dialog box. This property can be True for a document on any platform, but it only has an effect for PDF generated on the Macintosh.
PDFDestsMarked	F_Int	True if the book has documents with paragraphs or elements marked via MarkedForNamedDestination. One of two things must happen in order for this property to be True: The document was created in version 6.0 or later; the document was opened in version 6.0 or later, and the PDF FileSize Optimization client was run over it to mark all paragraphs or elements that are targets of hypertext links.
PDFDistillerAbsent	F_Int	A value of 1 indicates that there is no available Acrobat Distiller.
PDFDocInfo	Array of F_String	A list of strings expressing values to be set in the PDF Document Info dictionary when you save the document as PDF. Each dictionary entry is expressed as a pair of strings; the first string expresses the entry name, and the second string expresses the entry value.
PDFEndPage	F_String	The last page of the printing page range, in the FrameMaker numbering style.
PDFJobOption	F_String	The name of the Distiller Job Options. If the specified name does not exist in the Distiller Job Options list, then the first Distiller Job Option in the list is used.
PDFJobOptionsAbsent	F_Int	A value of 1 indicates that PDF Job Options are not available.
PDFOpenPage	F_String	The PDF page number, in the FrameMaker numbering style, at which Acrobat opens the generated PDF document.

Property:	Type:	Description:
PDFPageHeight	F_Metric	Page height for the generated PDF.
PDFPageWidth	F_Metric	Page width for the generated PDF.
PDFPrintPageRange	F_Int	True for generating PDF for the the specified page range; if False, FrameMaker generates PDF for the entire document or book.
PDFRegistrationMarks	F_Int	Registration marks for the generated PDF. May be one of: mFV_PDFRegistrationMarksNone mFV_PDFRegistrationMarksWestern mFV_PDFRegistrationMarksTombo
PDFSeparateFiles	F_Int	True, if a separate PDF file should be generated for each document in a book. This property can be set for single documents, but is ignored in that case.
PDFStartPage	F_String	The first page of the printing page range, in the FrameMaker numbering style
PDFStructure	F_Int	True if the document will create structured PDF when you save it as PDF. The structure is assigned via the PDFStructureLevel property of FMParagraphFormat objects.
PDFZoomFactor	F_Metric	When PDFZoomType is mFV_PDFZoomNone, the zoom percentage of the PDF document (metric 25% to 1600%. If the number is negative or zero, mFV_PDFZoomDefault takes effect.

Property:	Type:	Description:
PDFZoomType	F_Int	The PDF zoom setting with which Acrobat opens the generated PDF document. Can be one of: mFV_PDFZoomDefault mFV_PDFZoomPage mFV_PDFZoomWidth mFV_PDFZoomHeight mFV_PDFZoomNone If a different value is specified, mFV_PDFZoomDefault takes effect.

Contained Objects Properties

Property:	Type:	Description:
CurrentPage	FMPage	Current page (FMBodyPage, FMMasterPage, or FMReferencePage)
FirstAttrCondExprInDoc	FMAttrCondExpr	First FMAttrCondExpr in the document
FirstBodyPageInDoc	FMBodyPage	First FMBodyPage in the document
FirstCharFmtInDoc	FMCharacterFormat	First FMCharacterFormat in the list of the document's character formats
FirstColorInDoc	FMColor	First FMColor in the list of document's colours
FirstCombinedFontDefnInDoc	FMCombinedFontDefinition	First FMCombinedFontDefinition in the list of the document's combined font definitions
FirstCondFmtInDoc	FMConditionalFormat	First FMConditionalFormat in the list of the document's condition formats
FirstElementDefInDoc	FMElementDefinition	First FMElementDefinition in the lost of the document's element definitions
FirstFlowInDoc	FMFlow	First FMFlow in the list of the document's flows
FirstFmtChangeListInDoc	FMFormatChangeList	First FMFormatChangeList in the document's list of format change lists

Property:	Type:	Description:
FirstFnInDoc	FMFootnote	First FMFootnote in the list of the documents footnotes
FirstGraphicInDoc	FMGraphic	First graphic object in the list of the document's graphic objects (FMGraphic, or any child graphic objects)
FirstMarkerInDoc	FMMarker	First FMMarker in the list of the document's markers
FirstMarkerTypeInDoc	FMMarkerType	First FMMarkerType in the list of the document's marker types
FirstMasterPageInDoc	FMMasterPage	First FMMasterPage in the document
FirstPgfFmtInDoc	FMParagraphFormat	First FMParagraphFormat in the list of the document's paragraph tags
FirstPgfInDoc	FMParagraph	First FMParagraph in the list of the document's paragraphs
FirstRefPageInDoc	FMReferencePage	First FMReferencePage in the document
FirstRubiInDoc	FMRubi	First FMRubi composite in the list of the document's rubi composites
FirstRulingFmtInDoc	FMRulingFormat	First FMRulingFormat in the list of the document's ruling formats
FirstSelectedTiInDoc	FMTextInset	First text inset in the document's current selection (FMTextInset, FMTextInset_ApiClient, FMTextInset_Flow, FMTextInset_Text, or FMTextInset_TextTable)
FirstSelectedGraphicInDoc	FMGraphic	First selected graphic object in the list of selected graphic objects in the document (FMGraphic, or any child objects)
FirstTblFmtInDoc	FMTableFormat	First FMTableFormat in the list of the document's table formats
FirstTblInDoc	FMTable	First FMTable in the list of the document's tables

Property:	Type:	Description:
FirstTiInDoc	FMTextInset	First text inset in the list of the document's text insets (FMTextInset, FMTextInset_ApiClient, FMTextInset_Flow, FMTextInset_Text, or FMTextInset_TextTable)
FirstVarFmtInDoc	FMVariableFormat	First FMVariableFormat in the list of the document's variable formats
FirstVarInDoc	FMVariable	First FMVariable in the list of the document's variables
FirstXRefFmtInDoc	FMCrossReferenceFormat	First FMCrossReferenceFormat in the list of the document's cross-reference formats
FirstXRefInDoc	FMCrossReference	First FMCrossReference in the list of the document's cross-references
HiddenPage	FMHiddenPage	Hidden page (FMHiddenPage)
LastBodyPageInDoc	FMBodyPage	Last FMBodyPage in the document
LastMasterPageInDoc	FMMasterPage	Last FMMasterPage in the document
LastRefPageInDoc	FMReferencePage	Last FMReferencePage in the document
LeftMasterPage	FMMasterPage	Left master page (FMMasterPage)
MainFlowInDoc	FMFlow	Main flow (FMFlow)
NextOpenDocInSession	FMDocument	Next open document in the list of open documents in the session (FMDocument)
RightMasterPage	FMMasterPage	Right master page (FMMasterPage)
SelectedTable	FMTable	If any table cells are selected, the ID of the table containing them (FMTable)

Equation Properties

Property:	Type:	Description:
EqnIntegralSizeLarge	F_Metric	Point size of integral symbol in large equations (2 pt to 400 pt)
EqnIntegralSizeMed	F_Metric	Point size of integral symbol in medium equations (2 pt to 400 pt)

Property:	Type:	Description:
EqnIntegralSizeSmall	F_Metric	Point size of integral symbol in small equations (2 pt to 400 pt)
EqnLevel1SizeLarge	F_Metric	Point size of level 1 expression in large equations (2 pt to 400 pt)
EqnLevel1SizeMed	F_Metric	Point size of level 1 expression in medium equations (2 pt to 400 pt)
EqnLevel1SizeSmall	F_Metric	Point size of level 1 expression in small equations (2 pt to 400 pt)
EqnLevel2SizeLarge	F_Metric	Point size of level 2 expression in large equations (2 pt to 400 pt)
EqnLevel2SizeMed	F_Metric	Point size of level 2 expression in medium equations (2 pt to 400 pt)
EqnLevel2SizeSmall	F_Metric	Point size of level 2 expression in small equations (2 pt to 400 pt)
EqnLevel3SizeLarge	F_Metric	Point size of level 3 expression in large equations (2 pt to 400 pt)
EqnLevel3SizeMed	F_Metric	Point size of level 3 expression in medium equations (2 pt to 400 pt)
EqnLevel3SizeSmall	F_Metric	Point size of level 3 expression in small equations (2 pt to 400 pt)
EqnSigmaSizeLarge	F_Metric	Point size of sigma symbol in large equations (2 pt to 400 pt)
EqnSigmaSizeMed	F_Metric	Point size of sigma symbol in medium equations (2 pt to 400 pt)
EqnSigmaSizeSmall	F_Metric	Point size of sigma symbol in small equations (2 pt to 400 pt)
Functions	F_String	Character format tag of equation font to apply to Math Functions
Numbers	F_String	Character format tag of equation font to apply to Math Numbers
Symbols	F_String	Character format tag of equation font to apply to Math Symbols
SymbolsList	Array of F_String	List of math symbol fonts used in Equation Fonts dialog box

Property:	Type:	Description:
Variables	F_String	Character format tag of equation font to apply to Math Variables

Footnote Properties

Property:	Type:	Description:
FnCustNumString	F_String	Characters for custom document footnote numbers
FnFirstNum	F_Int	First document footnote number
FnFmt	F_String	Paragraph tag of footnote
FnHeightPerCol	F_Metric	Maximum height allowed for document footnotes (36 pt to 32767 pt)
FnInstancePosition	F_Int	Placement of document footnote number in footnote: mFV_FN_POS_SUPER: Superscript mFV_FN_POS_BASELINE: Baseline mFV_FN_POS_SUB: Subscript
FnInstancePrefix	F_String	Prefix to appear before document footnote number in footnote
FnInstanceSuffix	F_String	Suffix to appear after document footnote number in footnote
FnNumberingPerPage	F_Int	The document's footnote numbering type: mFV_NUM_CONTINUE: Continue the numbering from the previous file. mFV_NUM_RESTART: Restart numbering at the value specified by the associated FMDocument object's FP_FnFirstNum property. mFV_NUM_PER_PAGE: Restart numbering on each page.

Property:	Type:	Description:
FnNumComputeMethod	F_Int	Document footnote numbering style: mFV_FN_NUM_NUMERIC: Arabic mFV_FN_NUM_ROMAN_UC: Roman uppercase mFV_FN_NUM_ROMAN_LC: Roman lowercase mFV_FN_NUM_ALPHA_UC: Alphabetic uppercase mFV_FN_NUM_ALPHA_LC: Alphabetic lowercase mFV_FN_NUM_KANJI: Kanji characters mFV_FN_NUM_ZENKAKU: Zenkaku mFV_FN_NUM_ZENKAKU_UC: Zenkaku uppercase mFV_FN_NUM_ZENKAKU_LC: Zenkaku lowercase mFV_FN_NUM_KANJI_KAZU: Kazu mFV_FN_NUM_DAIJI: Daiji mFV_FN_NUM_CUSTOM: Custom numbering
FnNumStyle	F_Int	Position of footnote reference in document text: mFV_FN_POS_SUPER: Superscript mFV_FN_POS_BASELINE: Baseline mFV_FN_POS_SUB: Subscript
FnRefPosition	F_String	Prefix to appear before number in document text
FnRefPrefix	F_String	Suffix to appear after number in document text
FnRefSuffix	F_String	Characters for custom document footnote numbers

Hypertext Properties

FMDocument objects have the following properties that specify whether to parse and validate a hypertext command, and indicate the results of the parsing and validation. To

parse a hypertext command, set the value of HypertextCommandText to the command you want to parse. Setting the string executes the parser, and if HypertextDoValidate is true, setting the string executes validation as well. For more information, see "FMDocument Hypertext Properties" on page 403 of Appendix A, "Special Cases for Object Properties."

Property:	Type:	Description:
HypertextCommandText	F_String	The hypertext command to parse. Setting this value executes the parser. If HypertextDoValidate is True, the command will be parsed and validated.
HypertextDoValidate	F_Int	True if the next hypertext string sent to HypertextCommandText will be validated
HypertextParseBadParam	F_Int	If there was a parse error, an index into the HypertextParsedArgs string list.
HypertextParsedArgs	Array of F_String	The value of HypertextCommand, parsed into individual tokens
HypertextParsedClientName	F_String	For message commands, the name of the API client to receive the message.
HypertextParsedCmdCode	F_Int	The FrameMaker hypertext command in HypertextCommandText, as determined by the parser.
HypertextParsedCmdDest	F_Int	For link commands, the destination type in HypertextCommandText, as determined by the parser.
HypertextParsedCmdDestObjID	F_Int	For links to objects, the UID of the object in the target document.
HypertextParsedCmdDestObjType	F_Int	For links to objects, the type of the object in the target document.
HypertextParsedCmdMatrixColumns	F_Int	If HypertextParsedCmdCode is mFV_CmdMatrix, the number of columns in the matrix.
HypertextParsedCmdMatrixRows	F_Int	If HypertextParsedCmdCode is mFV_CmdMatrix, the number of rows in the matrix.
HypertextParsedDIFileName	F_String	For links to external files, the absolute path to the target file, expressed in platform independent syntax.

Property:	Type:	Description:
HypertextParsedFlowName	F_String	For popup and matrix commands, the name of the flow (on a reference page) that contains the popup or matrix list of commands.
HypertextParsedLinkName	F_String	For links to named targets, either the value of a newlink command, or a keyword such as FirstPage or LastPage.
HypertextParsedMessage	F_String	If HypertextParsedCmdCode is mFV_CmdAlert, mFV_CmdAlertTitle, or mFV_CmdMessage, the specified message for the hypertext command.
HypertextParsedPageName	F_String	For links to pages, the page number.
HypertextParsedTitle	F_String	If HypertextParsedCmdCode is mFV_CmdAlertTitle, the specified title for the alert box.
HypertextParseErr	F_Int	Non-zero if there was a parse error.
HypertextParseErrMsg	F_String	The message FrameMaker generates for a parse error
HypertextValidateErr	F_Int	Non-zero if HypertextDoValidate was true and there was a validation error.

General Document Properties

Property:	Type:	Description:
ChapNumComputeMethod	F_Int	The document's chapter numbering type: mFV_NUM_CONTINUE: Continue the numbering from the previous chapter. mFV_NUM_RESTART: Use the value specified for FP_ChapterNumber. mFV_NUM_SAME: Use the same chapter number as for the previous file.
ChapterNumber	F_Int	If ChapNumComputeMethod is mFV_NUM_RESTART, use this as the chapter number

Property:	Type:	Description:
ChapterNumStyle	F_Int	The numbering style; one of: mFV_NUMSTYLE_NUMERIC: Arabic. mFV_NUMSTYLE_ROMAN_UC: Roman numerals, uppercase. mFV_NUMSTYLE_ROMAN_LC: Roman, lowercase. mFV_NUMSTYLE_ALPHA_UC: Alphabetic, uppercase. mFV_NUMSTYLE_ALPHA_LC: Alphabetic, lowercase. mFV_NUMSTYLE_KANJI: Kanji. mFV_NUMSTYLE_ZENKAKU: Zenkaku. mFV_NUMSTYLE_ZENKAKU_UC: Zenkaku, uppercase. mFV_NUMSTYLE_ZENKAKU_LC: Zenkaku, lowercase. mFV_NUMSTYLE_KANJI_KAZU: Kazu. mFV_NUMSTYLE_DAIJI: Daiji. mFV_NUMSTYLE_TEXT: Text.
ChapterNumText	F_String	If ChapNumStyle is mFV_NUMSTYLE_TEXT, use this string as the chapter number.
Dictionary	Array of F_String	List of words to accept when spell-checking the document.
DocIsHelp	F_Int	True if the document is the FrameMaker product's Help document.
DocIsModified	F_Int	True if document has been modified. While this property is read-only, you can modify a document without setting this property to True by setting Untouchable to True for the document before your client modifies it.
DocIsViewOnly	F_Int	True if document is View Only.

Property:	Type:	Description:
DocOpenType	F_Int	Type of document the file was opened as: mFV_DOC_TYPE_BINARY: Frame binary document mFV_DOC_TYPE_TEXT: ASCII text document mFV_DOC_TYPE_MIF: MIF document mFV_DOC_TYPE_FILTER: a filtered document
DocSaveType	F_Int	Type of document the file is saved as: mFV_DOC_TYPE_BINARY: Frame binary document mFV_DOC_TYPE_TEXT: ASCII text document mFV_DOC_TYPE_MIF: MIF document mFV_DOC_TYPE_FILTER: a filtered document
DontUpdateTextInsets	F_Int	True if FrameMaker product doesn't automatically update text insets when it opens the document.
DontUpdateXRefs	F_Int	True if FrameMaker product doesn't automatically update cross-references when it opens or prints the document.
FormatOverride	F_Int	Specfies whether there are format overrides at the current insertion point. If the insertion point is in a text range that has a character format applied to it, FormatOverride is True if (and only if) the text formatting at the insertion point overrides the character format. If the insertion point is in a text range that has does not have a character format applied to it, FormatOverride is True if (and only if) the paragraph containing the insertion point has formatting that overrides the Paragraph Catalog format.
IsOnScreen	F_Int	True if document is visible on the screen.

Property:	Type:	Description:
MarkerTypeNames	Array of F_String	List of markertype names
Name	F_String	Filename of the document.
PageNumComputeMethod	F_Int	The document's paragraph numbering type: mFV_NUM_CONTINUE: Continue the numbering from the previous file. mFV_NUM_RESTART: Restart numbering at 1.
PgfNumComputeMethod	F_Int	The document's paragraph numbering type: mFV_NUM_CONTINUE: Continue the numbering from the previous file. mFV_NUM_RESTART: Restart numbering at 1.
StatusLine	F_String	String that appears in the document status bar. Note that this property always returns an empty string; it is effectively write-only If you set StatusLine to a string other than an empty string (""), the string will remain in the status bar until you reset it. To reset StatusLine so the FrameMaker product automatically updates the status line with normal status information, set it to an empty string ("").
TextSelection	udtTextRange	The currently selected text range or insertion point in the document.
Untouchable	F_Int	False by default. Setting this to True allows your client to modify a document without the FrameMaker product setting DocIsModified to True.
ViewOnlyWinPalette	F_Int	True if document acts like a palette when it is View Only.

Property:	Type:	Description:
VolNumComputeMethod	F_Int	The document's volume numbering type: mFV_NUM_CONTINUE: Continue the numbering from the previous volume. mFV_NUM_RESTART: Use the value specified for FP_VolumeNumber. mFV_NUM_SAME: Use the same volume number as for the previous file.
VolumeNumber	F_Int	If VolNumComputeMethod is mFV_NUM_RESTART, use this as the volume number
VolumeNumStyle	F_Int	The numbering style; one of: mFV_NUMSTYLE_NUMERIC: Arabic. mFV_NUMSTYLE_ROMAN_UC: Roman numerals, uppercase. mFV_NUMSTYLE_ROMAN_LC: Roman numerals,lowercase. mFV_NUMSTYLE_ALPHA_UC: Alphabetic, uppercase. mFV_NUMSTYLE_ALPHA_LC: Alphabetic, lowercase. mFV_NUMSTYLE_KANJI: Kanji. mFV_NUMSTYLE_ZENKAKU: Zenkaku. mFV_NUMSTYLE_ZENKAKU_UC: Zenkaku, uppercase. mFV_NUMSTYLE_ZENKAKU_LC: Zenkaku, lowercase. mFV_NUMSTYLE_KANJI_KAZU: Kazu. mFV_NUMSTYLE_DAIJI: Daiji. mFV_NUMSTYLE_TEXT: Text.
VolumeNumText	F_String	If VolNumStyle is mFV_NUMSTYLE_TEXT, use this string as the chapter number.

Changebar Properties

Property:	Type:	Description:
AutoChangeBars	F_Int	True if Automatic Change Bars is enabled
ChangeBarColor	FMColor	The spot color (FMColor)
ChangeBarDistance	F_Metric	Distance between change bar and text column
ChangeBarPosition	F_Int	Position of change bars: mFV_CB_COL_LEFT: Left of Column mFV_CB_COL_RIGHT: Right of Column mFV_CB_COL_NEAREST: Side Nearest to Page Edge mFV_CB_COL_FURTHEST: Side Farthest from Page Edge
ChangeBarThickness	F_Metric	Width of change bars

Condition Properties

Property:	Type:	Description:
ShowAll	F_Int	True if all conditions are displayed
ShowCondIndicators	F_Int	True if condition indicators (format overrides) are displayed

Menu Properties

Property:	Type:	Description:
MenuBar	FMMenu	The document's menu bar
ViewOnlyMenuBar	FMMenu	The document's menu bar when the document is locked

Page Properties

Property:	Type:	Description:
BottomMargin	F_Metric	Bottom page margin
ColGap	F_Metric	Size of gap between text columns

Property:	Type:	Description:
DocIsDoubleSided	F_Int	True if two-sided page layout
FirstPageNum	F_Int	Page number of first page
FirstPageVerso	F_Int	False for right first page; True for left first page
LeftMargin	F_Metric	Left page margin
NumCols	F_Int	Number of columns
PageHeight	F_Metric	Height of the document's pages. Setting this property automatically sets the PageHeight property of all of the document's body pages.
PageNumComputeMethod	F_Int	The document's page numbering type: mFV_NUM_CONTINUE: Continue the numbering from the previous file. mFV_NUM_RESTART: Restart numbering at the value specified by the associated FMDocument object's FirstPageNum property.

Property:	Type:	Description:
PageNumStyle	F_Int	Page numbering style: mFV_PAGE_NUM_NUMERIC: Arabic mFV_PAGE_NUM_ROMAN_UC: Roman uppercase mFV_PAGE_NUM_ROMAN_LC: Roman lowercase mFV_PAGE_NUM_ALPHA_UC: Alphabetic uppercase mFV_PAGE_NUM_ALPHA_LC: Alphabetic lowercase mFV_PAGE_NUM_KANJI: Kanji characters mFV_PAGE_NUM_ZENKAKU: Zenkaku mFV_PAGE_NUM_ZENKAKU_UC: Zenkaku uppercase mFV_PAGE_NUM_ZENKAKU_LC: Zenkaku lowercase mFV_PAGE_NUM_KANJI_KAZU: Kazu mFV_PAGE_NUM_DAIJI: Daiji
PageRounding	F_Int	How to round pages: mFV_PR_DEL_EMPTY: Delete Empty Pages mFV_PR_KEEP_NUM_EVEN: Make Page Count Even mFV_PR_KEEP_NUM_ODD: Make Page Count Odd mFV_PR_DONT_CHANGE: Don't Change Page Count
PageWidth	F_Metric	Width of the document's pages. Setting this property automatically sets the PageWidth property of all of the document's body pages.

Property:	Type:	Description:
PointPageNumStyle	F_Int	Point page numbering style: mFV_POINT_PAGE_NUM_NUMERIC: Arabic mFV_POINT_PAGE_NUM_ROMAN_UC: Roman uppercase mFV_POINT_PAGE_NUM_ROMAN_LC: Roman lowercase mFV_POINT_PAGE_NUM_ALPHA_UC: Alphabetic uppercase mFV_POINT_PAGE_NUM_ALPHA_LC: Alphabetic lowercase mFV_POINT_PAGE_NUM_KANJI: Kanji characters mFV_POINT_PAGE_NUM_ZENKAKU: Zenkaku mFV_POINT_PAGE_NUM_ZENKAKU_UC : Zenkaku uppercase mFV_POINT_PAGE_NUM_ZENKAKU_LC : Zenkaku lowercase mFV_POINT_PAGE_NUM_KANJI_KAZU: Kazu mFV_POINT_PAGE_NUM_DAIJI: Daiji
RightMargin	F_Metric	Right page margin
SmartQuotes	F_Int	True if Smart Quotes is enabled
SmartSpaces	F_Int	True if Smart Spaces is enabled
TopMargin	F_Metric	Top page margin

Print Properties

Property:	Type:	Description:
PrintBlankPages	F_Int	True if PageRounding allows empty page at end of document.
PrintCollated	F_Int	True if Collate is enabled.
PrintCols	F_Int	If PrintThumbnails is True, the number of columns to print.

Property:	Type:	Description:
PrintEmulsion	F_Int	Direction of print emulsion: mFV_EMUL_UP: Emulsion side up mFV_EMUL_DOWN: Emulsion side down
PrintEndPage	F_Int	Number of last page to print. Note that the value of FP_DocFluidFlow must be 0; you can't print a range of pages when a document is in fluid view.
PrintEndPageName	F_Int	Page number string for the last page to print; use this when the pages are numbered with a style other than mFV_PAGE_NUM_NUMERIC. Note that the value of FP_DocFluidFlow must be 0; you can't print a range of pages when a document is in fluid view.
PrintEndPoint	F_Int	Number of last point page to print.
PrinterName	F_String	Name of printer. Setting PrinterName on Windows has no effect. When you set PrinterName, you can set the printer to the default printer by specifying NULL.
PrintEvenPages	F_Int	True if Print Even-Numbered Pages is enabled.
PrintFileName	F_String	Filename of file to print to. When you set PrintFileName, you can set the filename to the default filename by specifying NULL. Setting this property on the Macintosh has no effect.
PrintImaging	F_Int	Type of print imaging: mFV_IMG_POSITIVE mFV_IMG_NEGATIVE
PrintLastSheetFirst	F_Int	True if Last Sheet First is enabled.
PrintLowRes	F_Int	True if Low-Resolution is enabled.
PrintManualFeed	F_Int	True if Manual Feed is enabled.
PrintNumCopies	F_Int	Number of copies to print.
PrintOddPages	F_Int	True if Odd-Numbered Pages is enabled.

Property:	Type:	Description:
PrintPaperHeight	F_Metric	Height of paper.
PrintPaperWidth	F_Metric	Width of paper.
PrintRegistrationMarks	F_Int	True if Registration Marks is enabled.
PrintRows	F_Int	If PrintThumbnails is True, the number of rows to print.
PrintScale	F_Int	Scale factor.
PrintScope	F_Int	Pages to print. Note that the value of DocFluidFlow must be 0; you can't print a range of pages when a document is in fluid view. mFV_PR_ALL: Print all pages mFV_PR_RANGE: Print a range of pages
PrintSeps	F_Int	True if Print Separations is enabled.
PrintStartPage	F_Int	Number of first page to print. Note that the value of DocFluidFlow must be 0; you can't print a range of pages when a document is in fluid view.
PrintStartPageName	F_Int	Page number string for the first page to print; use this when the pages are numbered with a style other than mFV_PAGE_NUM_NUMERIC. Note that the value of DocFluidFlow must be 0; you can't print a range of pages when a document is in fluid view.
PrintStartPoint	F_Int	Number of first point page to print.
PrintThumbnails	F_Int	True if Print Thumbnails is enabled.
PrintToFile	F_Int	True if Print Only to File is enabled.
SkipBlankSeps	F_Int	True if Skip Blank Separations is enabled (don't print blank color separations).
TrapwiseCompatibility	F_Int	True if Trapwise Compatibility is enabled. Setting this to True automatically sets FP_PrintToFile to True and PrintSep to False.

Rubi Properties

Property:	Type:	Description:
NarrowRubiSpaceForOther	F_Int	Allowable values are: mFV_Wide mFV_Narrow mFV_Proportional
RubiFixedSize	F_Metric	Fixed size for all rubi text (metric 2pts to 400pts). If this property and the FP_RubiSize property both have values, the most recently set property value is used.
RubiOverhang	F_Int	TRUE if rubi is allowed to overhang
RubiSize	F_Metric	Scaling factor for rubi text expressed as percentage of the current font size (metric 1% to 1000%). If this property and the FP_RubiFixedSize property both have values, the most recently set property value is used.
WideRubiSpaceForOther	F_Int	Allowable values are: mFV_Wide mFV_Narrow mFV_Proportional

Selection Properties

The following properties indicate what is currently selected in the document. A document can have text selected. For information on how FrameAC represents selected text, see "GetText" on page 298. For information on getting and setting the structural element selection in FrameMaker+SGML documents, see udtElementLoc in Chapter 6, "Data Type Reference."

Property:	Type:	Description:
ElementSelection	udtElementRange	The currently selected element range in the document.

Property:	Type:	Description:
FirstSelectedGraphicInDoc	FMGraphic	First selected graphic object in the document's list of selected graphic objects (FMGraphic, or any child graphic objects).
FirstSelectedTiInDoc	FMTextInset	First selected text inset in the list of selected text insets in the document (FMTextInset, FMTextInset_ApiClient, FMTextInset_Flow, FMTextInset_Text, or FMTextInset_TextTable).
SelectedTbl	FMTable	If any table cells are selected, the FMTable object of the table containing them.
TextSelection	udtTextRange	The currently selected text range or insertion point in the document.

Structure Properties

FMDocument objects have the following structure properties, which apply only to structured FrameMaker+SGML documents.

Property:	Type:	Description:
CustomElementList	Array of F_String	List of tags to display when ElementCatalogDisplay is set to mFV_ELCAT_CUSTOM.
DefaultExclusions	Array of F_String	List of exclusions inherited when document is included in a structured book.
DefaultInclusions	Array of F_String	List of inclusions inherited when document is included in a structured book.
ElementBoundaryDisplay	F_Int	Element boundary display options: mFV_ELEM_DISP_NONE: don't display any element boundaries mFV_ELEM_DISP_BRACKETS: display the bracketed boundaries mFV_ELEM_DISP_TAGS: display the element tags
ElementCatalog	Array of udtElementCatalogEntry	List of elements in Element Catalog.

Property:	Type:	Description:
ElementCatalogDisplay	F_Int	Catalog display options. Show tags for: mFV_ELCAT_STRICT: valid children for working start to finish mFV_ELCAT_LOOSE: valid children for working in any order mFV_ELCAT_CHILDREN: children allowed anywhere in parent mFV_ELCAT_ALL: all elements mFV_ELCAT_CUSTOM: the list of tags specified by CustomElementList
FirstElementDefInDoc	FMElementDefinition	ID of first element definition in the list of element definitions in the document (FMElementDefinition).
FirstFmtChangeListInDoc	FMFormatChangeList	ID of the first format change list in the list of format change lists in the document (FMFormatChangeList).
MaxBottomMargin	F_Metric	Maximum bottom margin allowed in the document.
MaxFirstIndent	F_Metric	Maximum first indent allowed in the document.
MaxFontSize	F_Metric	Maximum font size allowed in the document.
MaxLeading	F_Metric	Maximum leading allowed in the document.
MaxLeftIndent	F_Metric	Maximum left indent allowed in the document.
MaxLeftMargin	F_Metric	Maximum left margin allowed in the document.
MaxRightIndent	F_Metric	Maximum right indent allowed in the document.
MaxRightMargin	F_Metric	Maximum right margin allowed in the document.
MaxSpaceAbove	F_Metric	Maximum space above paragraph allowed in the document.

Property:	Type:	Description:
MaxSpaceBelow	F_Metric	Maximum space below paragraph allowed in the document.
MaxSpread	F_Metric	Obsolete property, but still functional. See corresponding "tracking" property below.
MaxStretch	F_Metric	Maximum character stretch (set width) expressed as a percentave of normal stretch for the font (metric –10% to 1000%).
MaxTabPosition	F_Metric	Maximum tab position allowed in the document.
MaxTopMargin	F_Metric	Maximum top margin allowed in the document.
MinBottomMargin	F_Metric	Minimum bottom margin allowed in the document.
MinFirstIndent	F_Metric	Minimum first indent allowed in the document.
MinFontSize	F_Metric	Minimum font size allowed in the document.
MinLeading	F_Metric	Minimum leading allowed in the document.
MinLeftIndent	F_Metric	Minimum left indent allowed in the document.
MinLeftMargin	F_Metric	Minimum left margin allowed in the document.
MinRightIndent	F_Metric	Minimum right indent allowed in the document.
MinRightMargin	F_Metric	Minimum right margin allowed in the document.
MinSpaceAbove	F_Metric	Minimum space above paragraph allowed in the document.
MinSpaceBelow	F_Metric	Minimum space below paragraph allowed in the document.
MinSpread	F_Metric	Obsolete property, but still functional. See corresponding "tracking" property below.

Property:	Type:	Description:
MinStretch	F_Metric	Minimum character stretch (set width) expressed as a percentage of normal stretch for the font (metric –10% to 1000%).
MinTabPosition	F_Metric	Minimum tab position allowed in the document.
MinTopMargin	F_Metric	Minimum top margin allowed in the document.
NewElemAttrDisplay	F_Int	Specifies attribute display properties for new elements: mFV_ATTR_DISP_NONE: don't display attributes mFV_ATTR_DISP_REQSPEC: display required and specified attributes mFV_ATTR_DISP_ALL: display all attributes
NewElemAttrEditing	F_Int	Specifies when the Edit Attributes dialog box appears for new elements: mFV_ATTR_EDIT_NONE mFV_ATTR_EDIT_REQUIRED mFV_ATTR_EDIT_ALWAYS
SeparateInclusions	F_Int	True if inclusions are listed separately in the Element Catalog.
StructuredApplication	F_String	The name of the structure application that is associated with the document. If the document has no associated structure application, this string is empty. For information on registering structured applications, see the online FrameMaker manual, *Structured Application Developer's Guide*.
UseInitialStructure	F_Int	True if FrameMaker+SGML inserts initial structure for new elements.
XmlDocType	F_String	The DOCTYPE parameter from the source XML.

Property:	Type:	Description:
XmlEncoding	F_String	The encoding parameter of the XML Declaration for the source XML document. The string is empty if no encoding is specified. If this property is set, the XML Declaration will contain the encoding parameter with this value on Save As XML. For information about the encoding that FrameMaker+SGML supports with a default installation, see the online FrameMaker manual, *Structured Application Developer's Guide*.
XmlFileEncoding	F_String	The encoding that was detected for the source XML book. If no encoding was specified for the source XML, FP_XmlEncoding will be an empty string. In that case, if this stringis set it will determine the encoding to use when saving as XML. If FP_XmlEncoding has a value, this string may be empty. For information about the encoding that FrameMaker+SGML supports with a default installation, see the online FrameMaker manual, *Structured Application Developer's Guide*.
XmlPublicId	F_String	The DOCTYPE public identifier for the source XML document.

Property:	Type:	Description:
XmlStandAlone	F_Int	An integer that specifies the XML standalone parameter for the XML document that was the source of this document. Can be one of: mFV_XML_STANDALONE_YES mFV_XML_STANDALONE_NO mFV_XML_STANDALONE_NODEC mFV_XML_STANDALONE_NONE The standalone parameter is declared in the XML Declaration. For a file with no XML Declaration, the value is mFV_XML_STANDALONE_NODEC. For an XML Declaration with no standalone parameter, this value is mFV_XML_STANDALONE_NONE.
XmlStyleSheet	F_String	The XML stylesheet processing instruction to write out to XML when saving the document as XML. The FDK does not verify that you use correct syntax in this string. The string you set should not include the PI delimiters, <? and ?>. For example, the string you supply for my.css would be: `"type=\"text\\css\" href=\"my.css\""` Only use this parameter to set a specific stylesheet specification. The parameter is always an empty string when you try to get the value. To get the list of stylesheet specifications associated with a document, use XmlStyleSheetList, below.

Property:	Type:	Description:
XmlStyleSheetList	Array of F_String	A list of stylesheet processing instructions for the current document. One document can have more than one stylesheet specification associated with it. The FDK does not verify that you use correct syntax in these strings. The strings should not include the PI delimiters, <? and ?>. For example, the string you supply for my.css would be: `"type=\"text\\css\" href=\"my.css\""` Setting a list to this parameter completely overwrites the preceeding list.
XmlSystemId	F_String	The DOCTYPE system identifier for the source XML document.
XmlUseBOM	F_Int	Indicates whether a byte order mark was detected when opening the source XML. Can be one of: mFV_XML_USEBOM_NA mFV_XML_USEBOM_NO mFV_XML_USEBOM_YES When saving as XML, it this is set to mFV_XML_USEBOM_YES, FrameMaker writes a byte order mark in the resulting XML.
XmlVersion	F_String	The XML Version that was specified in the XML Declaration when the file was opened. If no XML version was specified, F_ApiGetString() returns an empty string. If this string contains an invalid XML declaration, it will produce a parsing error when this document is saved as XML.
XmlWellFormed	F_Int	Indicates whether the source XML qualified as well formed. Can be one of: mFV_XML_WELLFORMED_NA mFV_XML_WELLFORMED_NO mFV_XML_WELLFORMED_YES

Table Footnote Properties

Property:	Type:	Description:
TblFnCellPosition	F_Int	Placement of footnote number in footnote text: mFV_FN_POS_SUPER: Superscript mFV_FN_POS_BASELINE: Baseline mFV_FN_POS_SUB: Subscript
TblFnCellPrefix	F_String	Prefix to appear before table footnote number in table cell
TblFnCellSuffix	F_String	Suffix to appear after table footnote number in table cell
TblFnCustNumString	F_String	Characters for custom table footnote numbers
TblFnFmt	F_String	Paragraph tag of table footnote
TblFnNumStyle	F_Int	Footnote numbering style for tables in document: mFV_FN_NUM_NUMERIC: Arabic mFV_FN_NUM_ROMAN_UC: Roman uppercase mFV_FN_NUM_ROMAN_LC: Roman lowercase mFV_FN_NUM_ALPHA_UC: Alphabetic uppercase mFV_FN_NUM_ALPHA_LC: Alphabetic lowercase mFV_FN_NUM_KANJI: Kanji characters mFV_FN_NUM_ZENKAKU: Zenkaku mFV_FN_NUM_ZENKAKU_UC: Zenkaku uppercase mFV_FN_NUM_ZENKAKU_LC: Zenkaku lowercase mFV_FN_NUM_KANJI_KAZU: Kazu mFV_FN_NUM_DAIJI: Daiji mFV_FN_NUM_CUSTOM: Custom numbering

Property:	Type:	Description:
TblFnPosition	F_Int	Placement of footnote number in text: mFV_FN_POS_SUPER: Superscript mFV_FN_POS_BASELINE: Baseline mFV_FN_POS_SUB: Subscript
TblFnPrefix	F_String	Prefix to appear before number in table footnote
TblFnSuffix	F_String	Suffix to appear after number in table footnote

Type-in Properties

The following properties specify how text will appear when typed in at the current location.

Property:	Type:	Description:
Capitalization	F_Int	Type of capitalization: mFV_CAPITAL_CASE_NORM mFV_CAPITAL_CASE_SMALL mFV_CAPITAL_CASE_LOWER mFV_CAPITAL_CASE_UPPER
ChangeBar	F_Int	True if Change Bars are enabled.
CharTag	F_String	Name of character format tag.
Color	FMColor	Spot color (FMColor).
CombinedFontFamily	FMCombinedFont Definition	Combined font definition (FMCombinedFontDefinition)
CondFmtIsShown	F_Int	True if condition is shown.
FontAngle	F_Int	Font angle (specifies an index into the array of font angles provided by the session property, FontAngleNames).
FontEncodingName	F_String	The font's encoding
FontFamily	F_Int	Font family (specifies an index into the array of font families provided by the session property, FP_FontFamilyNames).
FontPlatformName	F_String	Name that uniquely identifies a font on a specific platform. For more information, see "How FrameAC Represents Font Information" on page 417 in Appendix B, "FrameMaker Document and Session Architecture."
FontPostScriptName	F_String	Name given to a font when it is sent to a PostScript printer For more information, see "How FrameAC Represents Font Information" on page 417 in Appendix B, "FrameMaker Document and Session Architecture."
FontSize	F_Metric	Font size (2 pt to 400 pt).

Property:	Type:	Description:
FontVariation	F_Int	Font variation (specifies an index into the array of font variations provided by the session property FontVariationNames).
FontWeight	F_Int	Font weight (specifies an index into the array of font weights provided by the session property FontWeightNames).
InCond	Array of F_Int	Condition tags that apply to the text (array of FMConditionalFormat IDs).
KernX	F_Metric	Horizontal kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves a character right and a negative value moves a character left.
KernY	F_Metric	Vertical kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves characters up and a negative value moves characters down.
Overline	F_Int	True if Overline style is enabled.
PairKern	F_Int	True if Pair Kern is enabled.
Position	F_Int	Text position relative to baseline of text: mFV_POS_NORM: Normal mFV_POS_SUB: Subscript mFV_POS_SUPER: Superscript
SepOverride	FMColor	Custom color separation override (FMColor).
Spread	F_Metric	Obsolete property, but still functional. See corresponding "tracking" property below.
Stretch	F_Metric	Character stretch (set width) expressed as a percentage of normal stretch for the font (metric –10% to 1000%).
Strikethrough	F_Int	True if Strikethrough style is enabled.

Property:	Type:	Description:
StyleOverrides	F_Int	Style condition indicators for conditional text: mFV_CN_DOUBLE_UNDERLINE mFV_CN_NO_OVERRIDE mFV_CN_OVERLINE mFV_CN_SINGLE_UNDERLINE mFV_CN_STRIKETHROUGH
Underlining	F_Int	Type of underlining: mFV_CB_NO_UNDERLINE mFV_CB_SINGLE_UNDERLINE mFV_CB_DOUBLE_UNDERLINE mFV_CB_NUMERIC_UNDERLINE
UseSepOverride	F_Int	True if SepOverride overrides default.

Typographic Properties

Property:	Type:	Description:
LineBreakAfter	F_String	Characters at which it is permissible to break lines
SmallCapsSize	F_Metric	Scaling factor for small caps expressed as percentage of current font size (metric 1% to 1000%)
SubScriptShift	F_Metric	Baseline offset of subscripts expressed as percentage of current font size (metric 1% to 1000%)
SubScriptSize	F_Metric	Scaling factor for subscripts expressed as percentage of current font size (metric 1% to 1000%)
SubScriptStretch	F_Metric	Character stretch (set width) for subscripts expressed as a percentage of normal stretch for the font (metric –10% to 1000%).
SuperScriptShift	F_Metric	Baseline offset of superscripts expressed as percentage of current font size (metric 1% to 1000%)

Property:	Type:	Description:
SuperScriptSize	F_Metric	Scaling factor for superscripts expressed as percentage of the current font size (metric 1% to 1000%)
SuperScriptStretch	F_Metric	Character stretch (set width) for superscripts expressed as a percentage of normal stretch for the font (metric –10% to 1000%).

View Properties

Property:	Type:	Description:
IsIconified	F_Int	True if the document window is iconified.
IsInFront	F_Int	True if the document window is in front of other windows in the FrameMaker product session.
Label	F_String	The title in the document window title bar.
ScreenHeight	F_Int	Height of the document window in pixels.
ScreenWidth	F_Int	Width of the document window in pixels.
ScreenX	F_Int	The offset of the document window in pixels from the left side of the screen (or the left of the FrameMaker product application window on Windows). If you set a value that would result in the document window being off the screen, that value is ignored and the old value is retained.
ScreenY	F_Int	The offset of the document window in pixels from the top of the screen (or the top of the FrameMaker product application window on Windows). If you set a value that would result in the document window being off the screen, that value is ignored and the old value is retained.
SnapAngle	F_Metric	Angle of rotation for Snap Rotate.

Property:	Type:	Description:
SnapGridUnits	F_Metric	Units for Snap Grid Spacing (0 to 32768 pt).
SpotColorView	F_Int	Spot color separation view (0 to 6). 0 specifies View 1, 1 specifies View 2, and so on.
ViewBorders	F_Int	True if Borders is enabled.
ViewDisplayUnits	F_Metric	The F_Metric equivalent of one unit in the current Display Units. For example, if Display Units is points, this returns 65536.
ViewFontSizeUnits	F_Metric	The F_Metric equivalent of one unit in the current Font Size Unit. Font size units can be either Points or Q. If Points, this returns 65536. If Q, this returns 47098
ViewGrid	F_Int	True if View Grid is enabled.
ViewGridUnits	F_Metric	Units for Grid Lines.
ViewNoGraphics	F_Int	True if Graphics is not enabled.
ViewPageScrolling	F_Int	Page scrolling: mFV_SCROLL_VARIABLE mFV_SCROLL_HORIZONTAL mFV_SCROLL_VERTICAL mFV_SCROLL_FACING
ViewRulers	F_Int	True if Rulers is enabled.
ViewRulerUnits	F_Metric	Units for rulers display.
ViewTextSymbols	F_Int	True if Text Symbols is enabled.
Zoom	F_Metric	Zoom percentage of document (metric 25% to 1600%).
CurrentPage	FMPage	Current page (FMBodyPage, FMMasterPage, FMReferencePage).
IsOnScreen	F_Int	True if document is visible on the screen.

View-only Properties

Property:	Type:	Description:
DocFluidFlow	FMFlow	A FMFlow that is set to fluid view. To turn this off, set the value to 0.
DocIsViewOnly	F_Int	True if the document is a view-only document
ViewOnlyDeadCodes	Array of F_Int (unsigned)	F-codes that can't be executed in the document
ViewOnlyMenuBar	FMMenu	If the document has a specific menu bar, the FMMenu for that menu bar; otherwise no object
ViewOnlySelect	F_Int	Specifies whether user can select text or graphics in the document: mFV_VOS_USER_ONLY: the user can select text when pressing modifier keys, and link targets (cross-reference sources and newliniks) do not highlight mFV_VOS_NONE: the user can't select text, and links targets do not highlight mFV_VOS_YES: the user can select text (using modifier keys) and link targets are highlighted
ViewOnlyWinBorders	F_Int	True if the document has normal document borders; False if the document scroll bars and border buttons are suppressed
ViewOnlyWinMenubar	F_Int	True if the document has a document window menu bar
ViewOnlyWinPalette	F_Int	True if the document is a palette
ViewOnlyWinPopup	F_Int	True if the document window pop-up menu is available

Property:	Type:	Description:
ViewOnlyXRef	F_Int	Specifies the behavior of cross-references in the document: mFV_VOX_NOT_ACTIVE: cross-references are not active mFV_VOX_GOTO_BEHAVIOR: internal cross-references are active mFV_VOX_OPEN_BEHAVIOR: external cross-references are active mFV_VOX_ALERT: alert appears when cross-reference is clicked

Methods:

Method:	Description:
AddText	Adds text to the document at the passed location.
CenterOnText	Displays the document, centering on the specified text location.
Clear	Deletes the current selection from a document.
ClearAllChangebars	Clears all changebars from the document
Close	Closes the document.
Compare	Compares two documents.
Copy	Copies the current selection to the clipboard.
Cut	Cuts the current selection to the clipboard.
DeleteUndefinedAttribute	For a document, deletes all attributes that have no value assigned to them from the document.
DeleteText	Deletes the selected text.
DemoteElement	Demotes the selected element or elements.
ElementLocToTextLoc	Returns a text location that corresponds to the passed element location.
Find	Performs Search/Replace in the document.
GetElementCatalog	Gets the element catalog for the current document.
GetTextForRange	Gets text for the passed text range.
Import	Imports a graphic or text file into the document.

Methods: (Continued)

Method:	Description:
MergeIntoFirst	Merges the selected elements into the first element in the selection.
MergeIntoLast	Merges the selected elements into the last element in the selection.
NewAnchoredFormattedObject	Creates a new anchored, formatted object at the passed text location.
NewAnchoredObject	Creates a new anchored object at the passed text location.
NewElement	Creates a new element at the passed text location.
NewElementInHierarchy	Creates a new element at the passed element location.
NewBodyPage	Creates a new body page.
NewParagraph	Creates a new paragraph.
NewTable	Inserts a new table.
Paste	Pastes the clipboard contents at the current text selection.
PopClipboard	Pops the clipboard stack to the clipboard.
PromoteElement	Promotes the selected element or elements.
PushClipboard	Pushes the clipboard contents to the clipboard stack.
Redisplay	Updates the display of the document.
Reformat	Reformats the document—call this after disabling then enabling the document's Reformatting property.
Rehyphenate	Rehyphenates the document.
ResetEqnSettings	Resets equation settings to the default.
ResetReferenceFrames	After changing reference frames in the document, refreshes any references to these frames.
RestartPgfNumbering	Restarts paragraph numbering for the document.
Save	A scripted save for the document.
ScrollToText	Scrolls the document to the specified location.
Print	Prints the document.
ImportElementDefs	Imports an EDD into the document.
ImportFormats	Imports formats into the document.

Methods: (Continued)

Method:	Description:
SimpleSave	An unscripted save for the document.
SplitElement	Splits the selected element at the current insertion point.
TextLocToElementLoc	Converts a text location to a corresponding element location.
UnWrapElement	Removes the selected elements, but leaves their contents and child elements intact.
UpdateVariables	Updates the variables in the document.
UpdateXRefs	Updates the cross-references in the document.
WrapElement	Wraps the selected text and elements in the specified structural element.
NewNamedObject	Creates a new named object in the document—for example, a new character or paragraph format, a new master page, a new combined font definition, etc.
GetTextProps	Gets the text properties (such as format tag, font family and size, condition tag, etc.) for the passed text location.
SetTextProps	Sets the passed text properties (such as format tag, font family and size, condition tag, etc.) to the passed text range.

FMElement

Details

An FMElement object represents an instance of a structure element in a structured FrameMaker document.

See also

"FMObject" on page 166.

Properties

This reference manual divides the FMElement properties into the following groups:

- "Element General Properties" on page 130
- "Element CSS Property" on page 132
- "Element ID Properties" on page 133
- "Element Namespace Properties" on page 134

•"Element Validation Properties" on page 135

Element General Properties

Property:	Type:	Description:
AttrDisplay	F_Int	Specifies element's attribute display properties: mFV_ATTR_DISP_NONE: don't display attributes mFV_ATTR_DISP_REQSPEC: display required and specified attributes mFV_ATTR_DISP_ALL: display all attributes
Attributes	Array of udtAttribute	The element's attributes.
ContextLabel	F_String	The context label (if any) applied to the element.
ElementIsCollapsed	F_Int	True if element is collapsed in Structure View.

Property:	Type:	Description:
ElementType	F_Int	The type of element: mFV_FO_CONTAINER mFV_FO_TBL mFV_FO_MARKER mFV_FO_EQN mFV_FO_XREF mFV_FO_TBL_TITLE mFV_FO_TBL_HEADING mFV_FO_TBL_BODY mFV_FO_TBL_FOOTING mFV_FO_TBL_ROW mFV_FO_TBL_CELL mFV_FO_FOOTNOTE mFV_FO_GRAPHIC mFV_FO_SYS_VAR mFV_FO_RUBIGROUP mFV_FO_RUBI The following values, which were used in previous versions of FrameMaker+SGML, are no longer supported: mFV_FO_AFRAME mFV_FO_IMP_OBJECT mFV_FO_EMPTY mFV_FO_EMPTYPGF mFV_FO_VAR
FormatOverride	F_Int	True if the element has a format override.
ElementMarkedForNamed Destination	F_Int	Used for generatig PDF. If True, this element will have a corresponding Named Destination in the generated PDF.
MatchingFirstPgfClauses	Array of FMFormatRuleClause	Array of the first paragraph clauses (FMFormatRuleClause objects) in the element's definition that apply to the element.

Property:	Type:	Description:
MatchingLastPgfClauses	Array of FMFormatRuleClause	Array of the last paragraph clauses (FMFormatRuleClause objects) in the element's definition that apply to the element. The MatchingClauseTypeClauses properties specify only format rule clauses that are in the element definition's format rules (that is, the format rules specified by the element definition's TextFormatRules and ObjectFormatRules properties). Format rule clauses that the element inherits from ancestor elements may also apply to it. To determine whether an element inherits format rule clauses from ancestor elements, you must traverse up the structure tree and check the MatchingClauseTypeClauses properties for each ancestor element.
MatchingObjectClauses	Array of FMFormatRuleClause	The object clauses (FMFormatRuleClause object IDs) in the element's definition that apply to the element.
MatchingPrefixClauses	Array of FMFormatRuleClause	The prefix clauses (FMFormatRuleClause objects) in the element's definition that apply to the element.
MatchingSuffixClauses	Array of FMFormatRuleClause	The suffix clauses (FMFormatRuleClause objects) in the element's definition that apply to the element.
MatchingTextClauses	Array of FMFormatRuleClause	The text clauses (FMFormatRuleClause objects) in the element's definition that apply to the element.
TextRange	udtTextRange	Text range that the element encompasses (see the explanation below).
Unique	F_Int	The element's UID.
UserString	F_String	A string to which clients can store private data.

Element CSS Property

The FMElement object has the MatchesContextInUserString. To use MatchesContextInUserString, you first store a CSS context specification in the element's user string, then you check the value of this property. FrameMaker compares the user string to the element's current context, and sets MatchesContextInUserString to True or False according to the result. In this way, you can determine whether the current element matches the CSS context that you expect.

Another FDK client may utilize the user string—it's best practice to save the content of the user string to a temporary location, then restore it when you are done.

Property:	Type:	Description:
MatchesContextInUserString	F_Int	Compares the current content of the element's user string to the the element's current context. If this user string specifies a CSS context that matches the element's current context, F_ApiGetInt() returns True.

Element ID Properties

The following properties identify the IDs of objects that pertain to the current FMElement.

Property:	Type:	Description:
BookComponent	FMBookComponent	Component file in book (FMBookComponent).
ElementDef	FMElementDefinition	Element's element definition (FMElementDefinition).
FirstChildElement	FMElement	If element is a container, element's first child element (FMElement).
LastChildElement	FMElement	If element is a container, element's last child element (FMElement).
NextSiblingElement	FMElement	Element's next sibling element (FMElement).

Property:	Type:	Description:
Object	FMObject	ID of the object that an element contains. The type of object the ID specifies depends on the element definition as follows: mFV_FO_TBL: FMTable mFV_FO_MARKER: FMMarker mFV_FO_EQN: FMAnchoredFrame (containing the equation) mFV_FO_XREF: FMCrossReference mFV_FO_SYS_VAR: FMVariable mFV_FO_FOOTNOTE: FMFootnote mFV_FO_GRAPHIC: FMAnchoredFrame (containing the graphic) mFV_FO_TBL_TITLE: FMTable mFV_FO_TBL_HEADING: FMTable mFV_FO_TBL_BODY: FMTable mFV_FO_TBL_FOOTING: FMTable mFV_FO_TBL_ROW: FMRow mFV_FO_TBL_CELL: FMCell mFV_FO_RUBIGROUP: FMRubi mFV_FO_RUBI: FMRubi
ParentElement	FMElement	Element's parent element (FMElement).
PrevSiblingElement	FMElement	Element's previous sibling element (FMElement).

Element Namespace Properties

The following properties support XML Namespaces.

Property:	Type:	Description:

Namespace	Array of F_String	Prefix/path pairs defining namespaces for the element. This list must contain an even number of strings. For example:
		`prefix1, path1`
		`prefix2, path2`

NamespaceScope	FMElement	The FMElement which declares the namespace that is used to define the current element
NumNamespaces	F_Int	The number of namespaces declared in the current element. Readonly.

Element Validation Properties

Property:	Type:	Description:
AllowAsSpecialCase	F_Int	True if the element is treated as a special case.
AttributeValueInvalid	F_Int	True if the element contains an attribute value that is invalid.
BookComponentMissing	F_Int	True if a component file is missing from a book.
ContentIsLooselyValid	F_Int	True if the content is loosely valid (it has some missing elements).
ContentIsStrictlyValid	F_Int	True if the content of the element is strictly valid.
ContentMustBeEmpty	F_Int	True if the element can't have any content.
ContentNeededAtBegin	F_Int	True if content is needed at the beginning of the element.
ContentNeededAtEnd	F_Int	True if content is needed at end of the element. ContentNeededAtEnd is obsolete, but is supported for backward compatibility.
ElementIsExcludedInContext	F_Int	True if the element is excluded.

Property:	Type:	Description:
ElementIsInvalidInParent	F_Int	True if the element cannot occur anywhere in its current parent.
ElementIsInvalidInPosition	F_Int	True if the element is invalid in its current position.
ElementIsUndefined	F_Int	True if the element is undefined.
ErrorInBookComponent	F_Int	True if there is a validation error for a component in a book.
HoleBeforeElement	F_Int	True if there are one or more missing elements before the element within the same parent.
InvalidHighestLevel	F_Int	True if the element cannot be the highest-level element in the flow.
NextInvalidElement	FMElement	Next invalid element in the document.
TextIsInvalidInElement	F_Int	True if the element contains only text and the element definition disallows it. TextIsInvalidInElement is obsolete and is no longer supported.

Property:	Type:	Description:
ValidationFlags	F_Int	Bit flags specifying the element's validity. To determine all the ways in which an element is invalid without querying all the validation properties, query this property. Each bit flag in the returned value represents the value of the validation property with the corresponding name. For example, if the ElementTypeMismatch property is True, the mFV_ELEM_TYPE_MISMATCH flag is set. mFV_ELEM_UNDEFINED mFV_ELEM_TYPE_MISMATCH mFV_ELEM_EXCLUDED mFV_ELEM_INVALID_IN_PARENT mFV_ELEM_INVALID_AT_POSITION mFV_ELEM_HAS_TEXT_INVALID mFV_ELEM_CONTENT_MUST_BE_EMPTY mFV_ELEM_MISSING_CONTENT_BEFORE mFV_ELEM_MISSING_CONTENT_AT_BEG mFV_ELEM_MISSING_CONTENT_AT_END mFV_ELEM_NOT_VALID_AS_ROOT mFV_ELEM_BOOK_COMP_MISSING mFV_ELEM_BOOK_COMP_INVALID mFV_ELEM_ATTRVAL_REQUIRED mFV_ELEM_ATTRVAL_INVALID mFV_ELEM_CONTENT_STRICTLY_VALID mFV_ELEM_CONTENT_LOOSELY_VALID

Methods:

Method:	Description:
DeleteUndefinedAttribute	Deletes all attributes for the passed element that do not have any values.

FMElementDefinition

Details

FMElementDefinition objects represent the element definitions that appear in a document's Element Catalog.

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
AlsoInsert	Array of F_String	The list of the tags of child elements that are automatically inserted when an element is initially added.
AttributeDefs	Array of udtAttributeDef	The element definition's attribute definitions.
Comment	F_String	Text string of comment.

Property:	Type:	Description:
ElementDefType	F_Int	Type of formatter object represented by the element with element definition. mFV_FO_CONTAINER identifies a container element. Other values identify object (noncontainer) elements. mFV_FO_UNSPECIFIED mFV_FO_CONTAINER mFV_FO_TBL mFV_FO_MARKER mFV_FO_EQN mFV_FO_XREF mFV_FO_TBL_TITLE mFV_FO_TBL_HEADING mFV_FO_TBL_BODY mFV_FO_TBL_FOOTING mFV_FO_TBL_ROW mFV_FO_TBL_CELL mFV_FO_FOOTNOTE mFV_FO_GRAPHIC mFV_FO_SYS_VAR The following values, which were used in previous versions of FrameMaker+SGML, are no longer supported: mFV_FO_AFRAME mFV_FO_IMP_OBJECT mFV_FO_EMPTY mFV_FO_EMPTYPGF mFV_FO_VAR mFV_FO_RUBIGROUP mFV_FO_RUBI
ElementInCatalog	F_Int	True if the element is in the Element Catalog.
ElementPgfFormat	F_String	The name of the paragraph format applied to the element.

Property:	Type:	Description:
Exclusions	Array of F_String	List of excluded elements.
FirstPgfRules	Array of FMFormatRule	The objects of the first paragraph format rules (FMFormatRule objects).
GeneralRule	F_String	Text of the element's general rule.
GeneralRuleErrorOffsets	Array of F_Int	Contains the error offsets (two positions are specified only if the content rule is ambiguous).
Inclusions	Array of F_String	List of included elements.
InitStructurePattern	F_String	The initial structure pattern; for table elements, a comma delimited string that specifies the necessary child elements to automatically insert.
LastPgfRules	Array of FMFormatRule	The last paragraph format rules (FMFormatRule objects).
Name	F_String	Name of the element definition.
NextElementDefInDoc	FMElementDefinition	Next element definition in the document's list of element definitions (FMElementDefinition).
ObjectFmtRules	Array of FMFormatRule	The object format rules (FMFormatRule objects).
PrefixRules	Array of FMFormatRule	The prefix format rules (FMFormatRule objects).
SuffixRules	Array of FMFormatRule	The suffix format rules (FMFormatRule objects).
TableTagging	F_String	If the element is a table, the table format for new instances of the element.
TextFmtRules	Array of FMFormatRule	The text format rules (FMFormatRule objects).
ValidHighestLevel	F_Int	True if the element can be used as the highest-level element for a flow.

Methods:

Method:	Description:
ElementDefIsText <<IN THE IDL FILE - Not in Obj Browser>>	Some structural elements in FrameMaker documents are placeholders for text. For example, when a Para element contains text with a cross-reference element embedded in it, the ranges of text that surround the cross-reference element are treated as elements themselves. These elements are called text nodes. As you get an element's definition, you can use this method to check whether the definition is for a text node.
NewFormatRule	Adds a new format rule to the element definition.
DeleteUndefinedAttribute	Deletes all attributes with no value for elements of the type represented by the element definition.

FMEllipse

Details

FMEllipse objects represent drawn circles and ellipses in a document.

See also

"FMObject" on page 166, "FMGraphic" on page 155

Properties

Property:	Type:	Description:
RectangleIsSmoothed	F_Int	True if smoothing is enabled. This property is always True for FMEllipse objects.

FMFlow

Details

FMFlow objects represent the flows in a document. A flow is an ordered list of paragraphs, plus whatever text objects a paragraph can contain. To get the text in a flow, either get individual paragraphs in the flow and use the GetText method with those paragraphs, or use the GetText method with the entire flow.

FMFlow objects have the following properties.

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
FirstTextFrameInFlow	FMTextFrame	First text frame in flow.
FlowIsAutoConnect	F_Int	True if Autoconnect is enabled.
FlowIsFeathered	F_Int	True if Feather is enabled.
FlowIsPostScript	F_Int	True if flow is PostScript code.
FlowIsSynchronized	F_Int	True if Baseline Synchronization is enabled.
HighestLevelElement	FMElement	Highest-level element in flow.
LastTextFrameInFlow	FMTextFrame	Last text frame in flow.
MaxInterlinePadding	F_Metric	Maximum interline spacing.
MaxInterPgfPadding	F_Metric	Maximum interparagraph spacing.
MinHang	F_Metric	Maximum character height for synchronization of first line in column. If characters exceed this height, the FrameMaker product doesn't synchronize the first line.
Name	F_String	Name of flow tag.
NextFlowInDoc	FMFlow	Next flow in document.
SideHeadRoomInFlow	F_Int	True if Leave Room for Sideheads in Flow is enabled.
Spacing	F_Metric	Line spacing for synchronized baselines.

FMFootnote

Details

FMFootnote objects represent regular footnotes and table footnotes. FMDocument objects have properties that determine how footnotes appear. For more information, see "Footnote Properties" on page 97.

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
ContentHeight	F_Metric	The distance between the top of the footnote and the baseline of the last line in the footnote.
Element	FMElement	If the footnote is in a FrameMaker+SGML document, the FMElement object containing the footnote.
FirstPgf	FMParagraph	First paragraph in the footnote.
FnNum	F_Int	Footnote number.
InTextFrame	FMTextFrame	Text frame containing the footnote.
InTextObj	FMSubCol	Sub column that contains the footnote.
LastPgf	FMParagraph	Last paragraph in the footnote.
NextFn	FMFootnote	Next footnote in the text frame.
NextFnInDoc	FMFootnote	Next footnote in the document.
Overflowed	F_Int	True if the text in the footnote overflows
PrevFn	FMFootnote	Previous footnote in the text frame.
TextLoc	udtTextLoc	Text location of the footnote symbol.
Unique	F_Int	Footnote's UID.
Text	F_String	The text content of the footnote.

FMFormatChangeList

Details

FMFormatChangeList objects represent format change lists in structured FrameMaker documents.

See also

"FMObject" on page 166.

Format Change List Properties

Unlike other objects, FMFormatChangeList objects do not all have the same properties. All FMFormatChangeList objects have the properties listed in Format Change List General

Properties. However, each FMFormatChangeList object can have a different combination of the properties listed in the following sections.

This reference manual divides the FMElement properties into the following groups:

- "Format Change List General Properties" on page 144
- "Format Change List Advanced Properties" on page 144
- "Format Change List Asian Character Spacing Properties" on page 145
- "Format Change List Autonumbering Properties" on page 146
- "Format Change List Basic Properties" on page 146
- "Format Change List Font Properties" on page 148
- "Format Change List Pagination Properties" on page 151
- "Format Change List Table Cell Properties" on page 151

Format Change List General Properties

Property:	Type:	Description:
FmtChangeListInCatalog	F_Int	True if the format change list is in the Format Change List Catalog. False if it is in an element definition, as part of the text format rules.
Name	F_String	The name of the format change list if it is in the Format Change List Catalog.
NextFmtChangeListInDoc	FMFormatChangeList	The next format change list in the document.
PgfCatalogReference	F_String	A paragraph format tag if the format change list specifies one. If this property is set, you can't change any of the other format change list properties, except FP_Name.

Format Change List Advanced Properties

The following properties correspond with the advanced settings in the Paragraph Format dialog box.

Property:	Type:	Description:
AdjHyphens	F_Int	Number of allowable adjacent hyphens.
BottomSeparator	F_String	Name of frame to put below paragraph.
BottomSepAtIndent	F_Int	True if the position of the frame specified by `mFP_BottomSeparator` is at the current left indent.
Hyphenate	F_Int	True if Automatic Hyphenation is enabled.
HyphMinPrefix	F_Int	Minimum number of letters that must precede hyphen.
HyphMinSuffix	F_Int	Minimum number of letters that must follow a hyphen.
HyphMinWord	F_Int	Minimum length of a hyphenated word.
LetterSpace	F_Int	True if Word Spacing is enabled.
MaxSpace	F_Metric	Maximum word spacing (percentage of an em space in current font).
MinSpace	F_Metric	Minimum word spacing (percentage of an em space in current font).
OptSpace	F_Metric	Optimum word spacing.
TopSeparator	F_String	Name of frame to put above paragraph.
TopSepAtIndent	F_Int	True if the position of the frame specified by FP_TopSeparator is at the current left indent.

Format Change List Asian Character Spacing Properties

The following properties only pertain to characters in Asian font encodings.

Property:	Type:	Description:
MaxJLetSpace	F_Metric	Maximum Asian letter space.
MaxJRomSpace	F_Metric	Maximum Asian-Roman space.
MinJLetSpace	F_Metric	Minimum Asian letter space.
MinJRomSpace	F_Metric	Minimum Asian-Roman space.

Property:	Type:	Description:
OptJLetSpace	F_Metric	Optimum Asian letter space.
OptJRomSpace	F_Metric	Optimum Asian-Roman space.
YakumonoType	F_Int	The Yakumono rules to handle punctuation characters; can be one of mFV_FLOATING_YAKUMONO mFV_MONOSPACE_YAKUMONO mFV_FIXED_YAKUMONO

Format Change List Autonumbering Properties

The following properties correspond with the autonumbering settings in a paragraph format.

Property:	Type:	Description:
AutoNumChar	F_String	Character format for the automatic numbering string specified by FP_AutoNumString; "" if the default character format is used.
AutoNumString	F_String	Autonumber format string (for example, <n>.<n+>).
NumAtEnd	F_Int	True if numbering position is End of Paragraph; False if it is Beginning of Paragraph.
PgfIsAutoNum	F_Int	True if autonumbering is enabled.

Format Change List Basic Properties

The following properties correspond to the basic paragraph format settings.

Property:	Type:	Description:
FirstIndent	F_Metric	The paragraph's first-line left margin, measured from the left side of the current text column (0 cm to 100 cm).
FirstIndentChange	F_Metric	Amount by which to increase or decrease the first-line left margin.
FirstIndentIsRelative	F_Int	True if the first indent is relative to the left indent.

Property:	Type:	Description:
FirstIndentRelPos	F_Metric	Position relative to left indent if FirstIndentIsRelative is True.
Leading	F_Metric	Space below each line in the paragraph.
LeadingChange	F_Metric	Amount by which to increase or decrease the leading.
LeftIndent	F_Metric	The paragraph's left margin, measured from the left side of the current text column (0 cm to 100 cm).
LeftIndentChange	F_Metric	Amount by which to increase or decrease the left margin.
LineSpacingFixed	F_Int	True if the line spacing is fixed.
MoveTabs	F_Metric	Amount by which to move all tab positions in the paragraph.
NumTabs	F_Int	The number of tabs in the paragraph. To clear all the tabs in the paragraph, set FP_NumTabs to 0.
PgfAlignment	F_Int	Horizontal alignment of the paragraph: mFV_PGF_LEFT mFV_PGF_RIGHT mFV_PGF_CENTER mFV_PGF_JUSTIFIED
RightIndent	F_Metric	The paragraph's right margin, measured from the right side of the current text column.
RightIndentChange	F_Metric	Amount by which to increase or decrease the right margin.
SpaceAbove	F_Metric	The space above the paragraph.
SpaceAboveChange	F_Metric	Amount by which to increase or decrease the space above.
SpaceBelow	F_Metric	The space below the paragraph.
SpaceBelowChange	F_Metric	Amount by which to increase or decrease the space below.

Property:	Type:	Description:
Tabs	Array of udtTab	An array of tab descriptions that specify the positions and types of tab stops in the paragraph.

Format Change List Font Properties

The following properties correspond to the font settings in a paragraph format.

Property:	Type:	Description:
Capitalization	F_Int	Type of capitalization to use: mFV_CAPITAL_CASE_NORM mFV_CAPITAL_CASE_SMALL mFV_CAPITAL_CASE_LOWER mFV_CAPITAL_CASE_UPPER
ChangeBar	F_Int	True if Change Bars are on.
Color	FMColor	Spot color.
CombinedFontFamily	FMCombinedFont Definition	Combined font definition.
FontAngle	F_Int	Font angle (specifies an index into the array of font angles provided by the session property FP_FontAngleNames).
FontFamily	F_Int	Font family (specifies an index into the array of font families provided by the session property FP_FontFamilyNames).

Property:	Type:	Description:
Language	F_Int	Hyphenation and spell-checking language to use. mFV_LANG_BRAZILIAN mFV_LANG_BRITISH mFV_LANG_CANADIAN_FRENCH mFV_LANG_CATALAN mFV_LANG_DANISH mFV_LANG_DUTCH mFV_LANG_ENGLISH mFV_LANG_FINNISH mFV_LANG_FRENCH mFV_LANG_GERMAN mFV_LANG_ITALIAN mFV_LANG_NOLANGUAGE mFV_LANG_NORWEGIAN mFV_LANG_NYNORSK mFV_LANG_PORTUGUESE mFV_LANG_SPANISH mFV_LANG_SWEDISH mFV_LANG_SWISS_GERMAN mFV_LANG_JAPANESE mFV_LANG_TRADITIONAL_CHINESE mFV_LANG_SIMPLIFIED_CHINESE mFV_LANG_KOREAN
FontSize	F_Metric	Font size (2 pt to 400 pt).
FontSizeChange	F_Metric	Amount by which to increase or decrease the font size.
FontVariation	F_Int	Font variation (specifies an index into the array of font variations provided by the session property FP_FontVariationNames).

Property:	Type:	Description:
FontWeight	F_Int	Font weight (specifies an index into the array of font weights provided by the session property FP_FontWeightNames).
KernX	F_Metric	Horizontal kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves a character right and a negative value moves a character left.
KernY	F_Metric	Vertical kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves characters up and a negative value moves characters down.
Overline	F_Int	True if Overline is enabled.
PairKern	F_Int	True if Pair Kern is enabled.
Position	F_Int	Specifies position relative to baseline of text: mFV_POS_NORM: Normal mFV_POS_SUB: Subscript mFV_POS_SUPER: Superscript
Spread	F_Metric	Obsolete property, but still functional. See corresponding "tracking" property below.
SpreadChange	F_Metric	Obsolete property, but still functional. See corresponding "tracking" property below.
Stretch	F_Metric	Character stretch (set width) expressed as a percentage of normal stretch for the font (metric –10% to 1000%).
StretchChange	F_Metric	Amount expressed as a percentage (metric –10% to 1000%) by which to increase or decrease the character stretch.
Strikethrough	F_Int	True if Strikethrough is enabled.

Property:	Type:	Description:
Underlining	F_Int	Type of underlining: mFV_CB_NO_UNDERLINE mFV_CB_SINGLE_UNDERLINE mFV_CB_DOUBLE_UNDERLINE mFV_CB_NUMERIC_UNDERLINE

Format Change List Pagination Properties

The following properties correspond to the pagination settings for a paragraph format.

Property:	Type:	Description:
BlockLines	F_Int	The number of Widow/Orphan lines
KeepWithNext	F_Int	True if Keep With Next Paragraph is enabled
KeepWithPrev	F_Int	True if Keep With Previous Paragraph is enabled
Placement	F_Int	Paragraph placement: mFV_PGF_SIDEBODY mFV_PGF_SIDEHEAD_TOP mFV_PGF_SIDEHEAD_FIRST_BASELIN E mFV_PGF_SIDEHEAD_LAST_BASELINE mFV_PGF_RUN_IN mFV_PGF_STRADDLE mFV_PGF_STRADDLE_NORMAL_ONLY
RunInSeparator	F_String	String for Run-In Head Default Punctuation
Start	F_Int	Vertical placement of paragraph: mFV_PGF_ANYWHERE mFV_PGF_TOP_OF_COL mFV_PGF_TOP_OF_PAGE mFV_PGF_TOP_OF_LEFT_PAGE mFV_PGF_TOP_OF_RIGHT_PAGE

Format Change List Table Cell Properties

The folowing properties correspond to the table cell settings of a paragraph format.

Property:	Type:	Description:
CellBottomMargin	F_Metric	Amount added to default bottom margin of table cell.
CellBottomMarginChange	F_Metric	Amount by which to increase or decrease the cell bottom margin.
CellBottomMarginFixed	F_Int	True if the cell bottom margin is fixed.
CellLeftMargin	F_Metric	Amount added to default left margin of table cell.
CellLeftMarginChange	F_Metric	Amount by which to increase or decrease the cell left margin.
CellLeftMarginFixed	F_Int	True if the cell left margin is fixed.
CellRightMargin	F_Metric	Amount added to default right margin of table cell.
CellRightMarginChange	F_Metric	Amount by which to increase or decrease the cell right margin.
CellRightMarginFixed	F_Int	True if the cell right margin is fixed.
CellTopMargin	F_Metric	Amount added to default top margin of table cell.
CellTopMarginChange	F_Metric	Amount by which to increase or decrease the cell top margin.
CellTopMarginFixed	F_Int	True if the cell top margin is fixed.
CellVAlignment	F_Int	Vertical alignment of a paragraph when it is the first one in a cell: mFV_PGF_V_ALIGN_TOP mFV_PGF_V_ALIGN_MIDDLE mFV_PGF_V_ALIGN_BOTTOM

FMFormatRule

Details

The FMFormatRule object represents a format rule in a structured Framemaker document. One format rule is made up of format rule clauses—FrameAC uses FMFormatRuleClause objects to represent format rule clauses.

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
CountElements	Array of F_String	If the format rule is a level rule, the list of element tags to count among the element's ancestors; the tags are specified by the Count ancestors named element of the format rule.
ElementDef	FMElementDefinition	If the format rule is not nested, the FMElementDefinition that contains it.
FmtRuleClause	FMFormatRuleClause	If the format rule is nested, the FMFormatRuleClause that contains it.
FmtRuleClauses	Array of FMFormatRuleClause objects	The format rule clauses contained by this rule.
FmtRuleType	F_Int	The format rule's type: mFV_CONTEXT_RULE mFV_LEVEL_RULE
StopCountingAt	F_String	If the format rule is a level rule, the tag of the element at which to stop counting elements (the tag specified by the *Stop counting at first ancestor named* element.

Methods:

Method:	Description:
NewFormatRuleClause	Creates a FMFormatRuleClause in the format rule.

FMFormatRuleClause

Details

The FMFormatRuleClause object represents a single set of rule statements within a given FMFormatRule.

Properties

Property:	Type:	Description:
ContextLabel	F_String	The context label for generated files. It cannot contain white-space characters or any of these special characters: ( ) & | , * + ? < > % [ ] = ! ; : { } " When a user displays the Set Up dialog box to set up a generated file, the label appears next to elements to which the rule clause applies.
ElemPrefixSuffix	F_String	The text of the prefix or suffix. ElemPrefixSuffix specifies NULL if there is no prefix or suffix.
FmtChangeList	FMFormatChangeList	If the format rule clause specifies a format change list (RuleClauseType specifies mFV_RC_CHANGELIST), the object of the format change list.
FmtChangeListTag	F_String	If the format rule clause specifies a change list (RuleClauseType specifies mFV_RC_CHANGELIST_TAG), the change list's tag.
FmtRule	FMFormatRule	The format rule object containing the format rule clause.
FormatTag	F_String	The format tag if the format rule clause specifies one (RuleClauseType specifies mFV_RC_TAG). If IsTextRange is True, FormatTag specifies a character format tag; otherwise it specifies a paragraph tag, table tag, marker type, cross-reference format, or equation size.
IsTextRange	F_Int	True if the container element is formatted as a text range instead of a paragraph.
RuleClauseType	F_Int	The type of rule clause: mFV_RC_TAG mFV_RC_CHANGELIST_TAG mFV_RC_CHANGELIST mFV_RC_SUB_FMTRULE

Property:	Type:	Description:
Specification	F_String	The format clause's context or level specification.
SpecificationForCSS	Array of F_String	A list of CSS specifications that match the Specification for the current clause. For example, assume an element has specification of F < (G | H). This property returns the two following strings: G > F > E and H > F > E.
SubFmtRule	FMFormatRule	If the format rule clause contains a nested format rule (if RuleClauseType specifies mFV_RC_SUB_FMTRULE), the format rule object.

Methods:

Method:	Description:
NewFormatRule	Creates a FMFormatRule object that is nested within the current format rule clause.
NewFormatChangeList	Creates a FMFormatChangeList within the format rule clause.

FMGraphic

Details

FMGrabhic is the parent of the graphic objects that can be drawn in a document.

A FrameMaker document can contain a variety of graphic objects. Each of these graphic objects inherits the properties and methods of the FMGraphic object. Following is a list of the graphic objects that FrameMaker supports:

- FMArc
- FMEllipse
- FMGroup
- FMLine
- FMPageFrame
- FMPolyLine

- FMPolygon
- FMRectangle
- FMRoundRectangle
- FMTextFrame
- FMUnanchoredFrame

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
Angle	F_Metric	Angle of the object's rotation.
ArrowBaseAngle	F_Int	Arrowhead base angle in degrees.
ArrowLength	F_Metric	Arrowhead length (always rounded down to the nearest 1/256 point).
ArrowScaleFactor	F_Metric	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if FP_ArrowScaleHead is False.
ArrowScaleHead	F_Int	True if arrowhead is scaled as the line width changes.
ArrowTipAngle	F_Int	Arrowhead tip angle in degrees.
ArrowType	F_Int	Arrowhead style: mFV_ARROW_STICK mFV_ARROW_HOLLOW mFV_ARROW_FILLED
BorderWidth	F_Metric	Border width (0.015 pt to 360 pt).
Color	FMColor	The spot color.
Dash	Array of F_Metric	Dash style—see "Dash Patterns" on page 407 of Appendix A, "Special Cases for Object Properties."

Property:	Type:	Description:
Fill	F_Int	The fill pattern —numbers between 0 and 15; see “Pen and FIll properties” on page 407 of Appendix A, “Special Cases for Object Properties.” The FDK provides constants for the following fill patterns: mFV_FILL_BLACK mFV_FILL_WHITE mFV_FILL_CLEAR
FrameParent	FMGraphic	Frame containing the graphic object (FMAnchoredFrame or FMUnanchoredFrame).
GraphicCantBeSelected	F_Int	True if the graphic object can’t be selected.
GraphicIsSelected	F_Int	True if the graphic object is selected.
GroupParent	FMGroup	Group that the object is in. Anchored and unanchored frames do not have this property.
HeadArrow	F_Int	Arrowhead at end of line (True if line has arrowhead).
Height	F_Metric	Height of object (0.125 pt to 3600 pt).
LineCap	F_Int	Type of line end: mFV_CAP_BUTT mFV_CAP_ROUND mFV_CAP_SQUARE
LocX	F_Metric	Distance from the left side of the parent frame (–216 inches to 216 inches). If the graphic object is an anchored frame, the distance is calculated from the left side of the page frame. You can’t set FP_LocX for anchored frames.
LocY	F_Metric	Distance from the top of the parent frame (–216 inches to 216 inches). If the graphic object is an anchored frame, the distance is calculated from the top of the page frame. You can’t set FP_LocY for anchored frames.

Property:	Type:	Description:
NextGraphicInDoc	FMGraphic	Next graphic object in the document.
NextGraphicInFrame	FMGraphic	Next graphic object in the frame.
NextGraphicInGroup	FMGraphic	Next graphic object in the group.
NextSelectedGraphicInDoc	FMGraphic	Next selected graphic object in document.
ObjectAttributes	Array of F_String	A list of strings, each string expressing an attribute that is specified for an anchored frame in the Object Properties > Object Attributes dialog box. Each string is in the form of <tag>string_text</tag>. You can use the backslash to escape characters such as \n, \r, or \t for newline, carraige return, and tab, respectively. For a backslash character in the string text, type \\.
Overprint	F_Int	Specifies the overprint settings for the object mFV_OVERPRINT mFV_KNOCKOUT mFV_FROMCOLOR
Pen	F_Int	The pen pattern—numbers between 0 and 7; see “Pen and FIll properties” on page 407 of Appendix A, “Special Cases for Object Properties.” The FDK provides constants for several pen patterns: mFV_FILL_BLACK mFV_FILL_WHITE mFV_FILL_CLEAR
PrevGraphicInFrame	FMGraphic	Previous graphic object in the frame.
PrevGraphicInGroup	FMGraphic	Previous graphic object in the group.
RunaroundGap	F_Metric	If the object is a runaround object, the width of the runaround gap.
TailArrow	F_Int	Arrowhead at beginning of line (True if enabled).
TintPercent	F_Metric	The tint percentage

Property:	Type:	Description:
Unique	F_Int	The graphic object's UID.
Width	F_Metric	Width of object (0.125 pt to 3600 pt).

Methods:

Method:	Description:
NewChild	Creates a new graphic object such as a line, rectangle, ellipse, etc.

FMGroup

Details

Each FMGroup object represents a set of grouped graphic objects.

See also

"FMObject" on page 166, "FMGraphic" on page 155.

Properties

Property:	Type:	Description:
FirstGraphicInGroup	FMGraphic	First object in the group
LastGraphicInGroup	FMGraphic	Last object in the group

FMHiddenPage

Details

The FMHiddenPage is used to store hidden conditional text. If you want to manipulate hidden conditional text, you can either show all conditional text and manipulate it on the body pages, or hide a condition and manipulate the text on the hidden page.

See also

"FMObject" on page 166, "FMPage" on page 168.

Properties

Property:	Type:	Description:
Name	F_String	Name of hidden page

FMInset

Details

FMInset objects represent graphic files that are imported by reference.

See also

"FMObject" on page 166, "FMGraphic" on page 155.

Properties

Property:	Type:	Description:
ImportHint	F_String	Record identifying the filter used to import the graphic. The FrameMaker product uses this record to find the filter to use when updating the inset. For a complete description of the syntax of this string, see "Graphic Inset Import Hint Strings" on page 407 of Appendix A, "Special Cases for Object Properties."
InsetDpi	F_Int	Scaling information for bitmap file (corresponds to the value specified in the Image File Scaling Options dialog box when the graphics file is imported).
InsetEditor	F_String	Name of application to call to edit inset or imported object.
InsetFile	F_String	Platform-specific pathname if the inset is an external inset, or a null string (" ") if it is internal. The pathname can be document-relative.
InsetFileOrigName	F_String	Original name of the inset file.
InsetIsFixedSize	F_Int	True if scaling of bitmap file is inhibited.
InsetIsFlippedSideways	F_Int	True if inset is flipped about the vertical axis.
InsetUpdater	F_String	Not currently implemented.
PageNum	F_Int	For imported PDF, the page number of the PDF file to display in the inset. PDF page numbering begins with page 0.

FMLine

Details

FMLine objects represent lines drawn in a document.

See also

"FMObject" on page 166, "FMGraphic" on page 155.

Properties

Property:	Type:	Description:
NumPoints	F_Int	Number of vertices. The default is 2 (the line's start point and end point).
Points	Array of udtPoint	Array of x-y coordinate pairs that specify the line's vertices. The default coordinate pairs are for the line's start point and end point.

FMMarker

Details

FMMarker objects represent specific instances of markers in FrameMaker text.

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
Element	FMElement	If the marker is a structured marker in a FrameMaker+SGML document, the FMElement containing the marker.
MarkerText	F_String	The marker's text string.
MarkerType	FMMarkerType	The object of the current marker's FMMarkerType.
NextMarkerInDoc	FMMarker	Next FMMarker in the document.
TextLoc	udtTextLoc	Text location of the marker's symbol.
Unique	F_Int	The marker's UID.

FMMarkerType

Details

FMMarker types represent defined marker types. Marker types are defined and stored in the document, and they appear in the Marker Type list of the Marker dialog box. The FrameMaker product has a set of default marker types that you cannot delete.

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
InvariantName	F_String	An internal name for the marker type. By default, this is the same as the Name property. However, this can differ from Name if the user interface is in another language.
Name	F_String	The name of this marker type, as it appears in the user interface.
NextMarkerTypeInDoc	FMMarkerType	Next FMMarkerType in the document.
OldTypeNum	F_Int	A number to map markers from documents earlier than version 5.5 to this marker type. For example, assume the name of a marker type is MyMarkerType, and the OldTypeNum is 11. Then markers of type 11 from earlier documents will import as markers of type MyMarkerType.
Public	F_Int	True if the marker type should appear in the user interface. The default is True.
Required	F_Int	True if the marker type is required by the FrameMaker product. The default is False.
Transient	F_Int	True if markers of this type shoud not be saved to files. The default is False.

FMMasterPage

Details

FMMasterPage objects represent the master pages in a document.

See also

"FMObject" on page 166, "FMPage" on page 168.

Properties

Property:	Type:	Description:
Name	F_String	Name of master page (for example, Right or Left)
PageNum	F_Int	Page number

FMMath

Details

FMMath objects represent equations in documents. The MathFullForm property corresponds to the MIF <MathFullForm> statement that defines the mathematical structure of an equation. Each expression defines a component of the equation and can be nested within other expressions, as in string1[string2].

For example, to create the equation x > y, you specify the following string for the FMMath object's MathFullForm property:

```
greaterthan[char[x],char[y]]
```

For more information on the MIF <MathFullForm> statement, see the online FrameMaker manual, *Mif Reference*.

FMDocument objects have properties that specify how all the equations in a document appear. For a list of these properties, see "Equation Properties" on page 95.

See also

"FMObject" on page 166, "FMGraphic" on page 155.

Properties

Property:	Type:	Description:
BasePointX	F_Metric	Horizontal placement of text line base point relative to left side of frame
BasePointY	F_Metric	Vertical placement of text line base point relative to top of frame
MathFullForm	F_String	String representing the expression

Property:	Type:	Description:
MathSize	F_Int	Equation size: mFV_MATH_LARGE mFV_MATH_MEDIUM mFV_MATH_SMALL
TextLineType	F_Int	Type of text line: mFV_TEXTLINE_LEFT mFV_TEXTLINE_RIGHT mFV_TEXTLINE_CENTER mFV_TEXTLINE_MATHD

FMMenu

Details

The FMMenu is a menu which can contain FMCommand or FMMenuItemSeperator objects. One FMMenu object can contain other FMMenu objects—this is how you create submenus (or cascading menus).

See also

"FMObject" on page 166, "FMCommandObject" on page 84.

Properties

Property:	Type:	Description:
FirstMenuItemInMenu	FMCommand	The first FMCommand or FMMenu in the menu.
MenuType	F_Int	Type of menu: mFV_MENU_MENUBAR: a menu bar defined by the FrameMaker product mFV_MENU_POPUP: a pop-up menu mFV_MENU_ADHOCRULER: an ad hoc formatting menu that appears on the ruler mFV_MENU_DEFAULT: a pull-down or pull-right menu

Methods:

Method:	Description:
AddMenuToMenu	Adds a new FMMenu to the current menu object.

FMMenuItemSeperator

Details

The FMMenuItemSeperator is a special FMCommandObject that is contained by a FMMenu. The separator has no properties except those inherited by the FMCommandObject.

FMMenuItemSeperator objects use predefined names, which are `!Separator`, `Separator1`, `Separator2`, `Separator3`, `Separator4`, and `Separator5`.

See also

"FMObject" on page 166, "FMCommandObject" on page 84.

FMObject

Details

FMObject is a generic object that is the ancestor to all other FrameAC objects. The properties and methods of FMObject are inherited by all FrameAC objects.

Properties

Property:	Type:	Description:
Id	F_Int (Obj ID)	An integer to identify this object. The ID value only persists for the current document session—closing the document and opening it makes previous ID values unreliable. For FMSession, this value is mFV_SessionId.
DocId	F_Int (Obj ID)	An integer to identify the document that contains the object. You can use this integer to refer to the document from the current FMSession.
UserString	F_String	A string that you can store with the object. Note that other plugins may rely on this string—recommended use is to store the current value to a variable, use this to store your own value, then restore the original value when you're done.

Methods:

Method:	Description:
Delete	Deletes the object from its container object. For example, deletes a paragraph from a flow or document.
GetText	Gets the text contained by the object. For example, you can get text for a flow, a paragraph, a table cell, or other objects that contain text.
ObjectValid	Determines whether the current object has been successfully created.
GetPropVals	Returns a list of properties for the current object.
SetPropVals	Sets a list of properties to the current object.

FMPage

Details

FMPage is an object that is the ancestor of all FMBodyPage, FMHiddenPage, FMMasterPage, and FMReferencePage objects. These page-type objects inherit the properties and methods of FMPage.

See also

"FMObject" on page 166, .

Properties

Property:	Type:	Description:
PageFrame	FMPageFrame	The page's page frame (FMPageFrame).
PageHeight	F_Metric	Height of page.
PageNext	FMPage	Next page in the document.
PagePrev	FMPage	Previous page in the document.
PageWidth	F_Metric	Width of page.

Methods:

Method:	Description:
ApplyPageLayout	Copies the layout from one FMPage and applies it to the current FMPage.

FMPageFrame

Details

FMPageFrame objects represent the frame on a page that contains all other graphic objects on a page. To arrive at the FMPageFrame, you can start from any graphic object and repeatedly get the object's FrameParent property. The last valid FrameParent you get will be the FMPageFrame.

FMPageFrame objects are effectively identical to FMUnanchoredFrame objects.

See also

"FMObject" on page 166, "FMGraphic" on page 155.

3

FMParagraph

Details

FMParagraph objects represent the paragraphs in a document flow. Note that FMTextLine objects cannot contain FMParagraph objects. FrameAC uses FMParagraphFormat objects to represent the paragraph formats that are stored in a document.

To get the text of a paragraph, use the FMParagraph object's GetText method.

See also

"FMObject" on page 166.

Properties

This reference manual divides the FMParagraph properties into the following groups:

- "FMParagraph Asian Character Spacing Properties" on page 169
- "FMParagraph Autonumbering Properties" on page 170
- "FMParagraph Default Font Properties" on page 171
- "FMParagraph Table Cell Properties" on page 173
- "FMParagraph Hyphenation Properties" on page 174
- "FMParagraph Linguistic Properties" on page 174
- "FMParagraph General Properties" on page 176
- "FMParagraph Indent Properties" on page 176
- "FMParagraph Line Spacing Properties" on page 176
- "FMParagraph Placement Properties" on page 177
- "FMParagraph Object Pointer Properties" on page 178
- "FMParagraph Reference Frame Properties" on page 178
- "FMParagraph Tab Properties" on page 178
- "FMParagraph Tagging Properties" on page 178
- "FMParagraph Word Spacing Properties" on page 179

FMParagraph Asian Character Spacing Properties

Property:	Type:	Description:
MaxJLetSpace	F_Metric	Maximum Asian letter space
MaxJRomSpace	F_Metric	Maximum Asian-Roman space

Property:	Type:	Description:
MinJLetSpace	F_Metric	Minimum Asian letter space
MinJRomSpace	F_Metric	Minimum Asian-Roman space
OptJLetSpace	F_Metric	Optimum Asian letter space
OptJRomSpace	F_Metric	Optimum Asian-Roman space
YakumonoType	F_Int	The Yakumono rules to handle punctuation characters; can be one of mFV_FLOATING_YAKUMONO mFV_MONOSPACE_YAKUMONO mFV_FIXED_YAKUMONO

FMParagraph Autonumbering Properties

Property:	Type:	Description:
AutoNumChar	F_String	Character Format for the automatic numbering string specified by utoNumString; "" if the default character format is used
AutoNumString	F_String	Autonumber Format string; for example, "<n>.<n+>"
NumAtEnd	F_Int	True if numbering position is End of Paragraph; False if it is Beginning of Paragraph
PgfIsAutoNum	F_Int	True if autonumbering is enabled
PgfNumber	F_String	The formatted string representation of the paragraph number; for example, 1.2 for a paragraph whose AutoNumString property is set to <n>.<n+>

FMParagraph Default Font Properties

Property:	Type:	Description:
Capitalization	F_Int	Type of capitalization to use: mFV_CAPITAL_CASE_NORM mFV_CAPITAL_CASE_SMALL mFV_CAPITAL_CASE_LOWER mFV_CAPITAL_CASE_UPPER
ChangeBar	F_Int	True if Change Bars are on.
Color	FMColor	Spot color.
CombinedFontFamily	FMCombinedFont Definition	Combined font definition.
FontAngle	F_Int	Font angle (specifies an index into the array of font angles provided by the session property FontAngleNames).
FontEncodingName	F_String	The font's encoding
FontFamily	F_Int	Font family (specifies an index into the array of font families provided by the session property FontFamilyNames).
FontPlatformName	F_String	Name that uniquely identifies a font on a specific platform. For more information, see "How FrameAC Represents Font Information" on page 417 in Appendix B, "FrameMaker Document and Session Architecture."
FontPostScriptName	F_String	Name given to a font when it is sent to a PostScript printer. For more information, see "How FrameAC Represents Font Information" on page 417 in Appendix B, "FrameMaker Document and Session Architecture."
FontSize	F_Metric	Font size (2 pt to 400 pt).
FontVariation	F_Int	Font variation (specifies an index into the array of font variations provided by the FMSession property FontVariationNames).

Property:	Type:	Description:
FontWeight	F_Int	Font weight (specifies an index into the array of font weights provided by the FMSession property FontWeightNames).
KernX	F_Metric	Horizontal kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves a character right and a negative value moves a character left.
KernY	F_Metric	Vertical kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves characters up and a negative value moves characters down.
Overline	F_Int	True if Overline is enabled.
PairKern	F_Int	True if Pair Kern is enabled.
Position	F_Int	Specifies position relative to baseline of text: mFV_POS_NORM: Normal mFV_POS_SUB: Subscript mFV_POS_SUPER: Superscript
Spread	F_Metric	Obsolete property, but still functional. See corresponding "tracking" property below.
Stretch	F_Metric	Character stretch (set width) expressed as a percentage of normal stretch for the font (metric –10% to 1000%).
Strikethrough	F_Int	True if Strikethrough is enabled.
Underlining	F_Int	Type of underlining: mFV_CB_NO_UNDERLINE mFV_CB_SINGLE_UNDERLINE mFV_CB_DOUBLE_UNDERLINE mFV_CB_NUMERIC_UNDERLINE

FMParagraph Table Cell Properties

Property:	Type:	Description:
CellBottomMargin	F_Metric	Amount added to default bottom margin of table cell.
CellLeftMargin	F_Metric	Amount added to default left margin of table cell.
CellMarginsFixed	F_Int	Specifies which cell margins are fixed. The following values can be ORed into it: mFV_PGF_FIXED_B_MARGIN: the bottom margin is fixed mFV_PGF_FIXED_L_MARGIN: the left margin is fixed mFV_PGF_FIXED_R_MARGIN: the right margin is fixed mFV_PGF_FIXED_T_MARGIN: the top margin is fixed If the margin for a cell is fixed, the margin property specifies the absolute value of the cell margin. For example, if mFV_PGF_FIXED_B_MARGIN is set, CellBottomMargin specifies the absolute value of the cell's bottom margin, overriding the cell margin specified by the table format. If mFV_PGF_FIXED_B_MARGIN is not set, CellBottomMargin is added to the margin specified by the table format.
CellRightMargin	F_Metric	Amount added to default right margin of table cell.
CellTopMargin	F_Metric	Amount added to default top margin of table cell.
CellVAlignment	F_Int	Vertical alignment of a paragraph when it is the first one in a cell: mFV_PGF_V_ALIGN_TOP mFV_PGF_V_ALIGN_MIDDLE mFV_PGF_V_ALIGN_BOTTOM

FMParagraph Hyphenation Properties

Property:	Type:	Description:
AdjHyphens	F_Int	Number of allowable adjacent hyphens.
Hyphenate	F_Int	True if Automatic Hyphenation is enabled.
HyphMinPrefix	F_Int	Minimum number of letters that must precede hyphen.
HyphMinSuffix	F_Int	Minimum number of letters that must follow a hyphen.
HyphMinWord	F_Int	Minimum length of a hyphenated word.

FMParagraph Linguistic Properties

The following properties describe the paragraph's language, and identify whether it has been checked by the spell checker.

Property:	Type:	Description:
Language	F_Int	Hyphenation and spell-checking language to use: mFV_LANG_BRAZILIAN mFV_LANG_BRITISH mFV_LANG_CANADIAN_FRENCH mFV_LANG_CATALAN mFV_LANG_DANISH mFV_LANG_DUTCH mFV_LANG_ENGLISH mFV_LANG_FINNISH mFV_LANG_FRENCH mFV_LANG_GERMAN mFV_LANG_ITALIAN mFV_LANG_NOLANGUAGE mFV_LANG_NORWEGIAN mFV_LANG_NYNORSK mFV_LANG_PORTUGUESE mFV_LANG_SPANISH mFV_LANG_SWEDISH mFV_LANG_SWISS_GERMAN mFV_LANG_JAPANESE mFV_LANG_TRADITIONAL_CHINESE mFV_LANG_SIMPLIFIED_CHINESE mFV_LANG_KOREAN
PgfSpellChecked	F_Int	True if paragraph has been spell-checked.

FMParagraph General Properties

Property:	Type:	Description:
FormatOverride	F_Int	True if the paragraph is part of a text inset that retains formatting information from the source document. The paragraph is not affected by global formatting performed on the document.
Locked	F_Int	True if the paragraph contains a paragraph format override.
Unique	F_Int	The paragraph's UID.

FMParagraph Indent Properties

Property:	Type:	Description:
FirstIndent	F_Metric	First-line left margin, measured from left side of current text column (0 cm to 100 cm).
LeftIndent	F_Metric	Left margin, measured from left side of current text column (0 cm to 100 cm).
RightIndent	F_Metric	Right margin, measured from right side of current text column.

FMParagraph Line Spacing Properties

Property:	Type:	Description:
Leading	F_Metric	Space below each line in a paragraph
LineSpacing	F_Int	Space between lines in a paragraph measured from baseline to baseline: mFV_PGF_FIXED: default font size mFV_PGF_PROPORTIONAL: largest font in line mFV_PGF_FLOATING: largest ascender in line

FMParagraph Placement Properties

Property:	Type:	Description:
BlockLines	F_Int	The number of Widow/Orphan lines.
KeepWithNext	F_Int	True if Keep With Next Paragraph is enabled.
KeepWithPrev	F_Int	True if Keep With Previous Paragraph is enabled.
PgfAlignment	F_Int	Horizontal alignment of paragraph: mFV_PGF_LEFT mFV_PGF_RIGHT mFV_PGF_CENTER mFV_PGF_JUSTIFIED
Placement	F_Int	Paragraph placement: mFV_PGF_SIDEBODY mFV_PGF_SIDEHEAD_TOP mFV_PGF_SIDEHEAD_FIRST_BASELIN E mFV_PGF_SIDEHEAD_LAST_BASELINE mFV_PGF_RUN_IN mFV_PGF_STRADDLE mFV_PGF_STRADDLE_NORMAL_ONLY
RunInSeparator	F_String	String for Run-In Head Default Punctuation.
SpaceAbove	F_Metric	Space above paragraph.
SpaceBelow	F_Metric	Space below paragraph.
Start	F_Int	Vertical placement of paragraph: mFV_PGF_ANYWHERE mFV_PGF_TOP_OF_COL mFV_PGF_TOP_OF_PAGE mFV_PGF_TOP_OF_LEFT_PAGE mFV_PGF_TOP_OF_RIGHT_PAGE

FMParagraph Object Pointer Properties

Property:	Type:	Description:
InTextFrame	FMTextFrame	Text frame containing the paragraph.
InTextObj	FMSubCol or FMFootnoteor FMCell	Subcolumn, footnote, or table cell the paragraph begins.
NextPgfInDoc	FMParagraph	Next FMParagraph in the document.
NextPgfInFlow	FMParagraph	Next FMParagraph in the flow.
PrevPgfInFlow	FMParagraph	Previous FMParagraph in the flow.

FMParagraph Reference Frame Properties

Property:	Type:	Description:
BottomSeparator	F_String	Name of frame to put below paragraph.
TopSeparator	F_String	Name of frame to put above paragraph.

FMParagraph Tab Properties

The following properties specify the tab settings in the paragraph (not tab characters).

Property:	Type:	Description:
NumTabs	F_Int	Number of tabs in the paragraph.
Tabs	Array of udtTab	Array of tab descriptions that specify the positions and types of tab stops.

FMParagraph Tagging Properties

Property:	Type:	Description:
Name	F_String	Name of paragraph format.
NextTag	F_String	Tag for new next paragraph.
UseNextTag	F_Int	True if Next Paragraph Tag is enabled.

FMParagraph Word Spacing Properties

Property:	Type:	Description:
LetterSpace	F_Int	True if Word Spacing is enabled.
MaxSpace	F_Metric	Maximum word spacing (percentage of an em space in current font).
MinSpace	F_Metric	Minimum word spacing (percentage of an em space in current font).
OptSpace	F_Metric	Optimum word spacing.

FMParagraphFormat

Details

FMParagraphFormat objects represent the paragraph formats that are defined in a document's paragraph format catalog. Instances of individual paragraphs are represented by FMParagraph objects.

See also

"FMObject" on page 166.

Properties

This reference manual divides the FMParagraphFormat properties into the following groups:

- "FMParagraphFormat Acrobat Properties" on page 180
- "FMParagraphFormat Asian Character Spacing Properties" on page 180
- "FMParagraphFormat Autonumbering Properties" on page 181
- "FMParagraphFormat Default Font Properties" on page 181
- "FMParagraphFormat Table Cell Properties" on page 183
- "FMParagraphFormat Hyphenation Properties" on page 184
- "FMParagraphFormat Linguistic Properties" on page 185
- "FMParagraphFormat Indent Properties" on page 186
- "FMParagraphFormat Line Spacing Properties" on page 187
- "FMParagraphFormat Placement Properties" on page 187
- "FMParagraphFormat Object Pointer Properties" on page 188
- "FMParagraphFormat Reference Frame Properties" on page 188
- "FMParagraphFormat Tab Properties" on page 188

•"FMParagraphFormat Tagging Properties" on page 189
•"FMParagraphFormat Word Spacing Properties" on page 189

FMParagraphFormat Acrobat Properties

Property:	Type:	Description:
AcrobatLevel	F_Int	Retained in Version 6.0 or later for backward compatibility. Use FP_PDFLevel instead.
PDFStructureLevel	F_Int	The PDF structure level of paragraphs with the current format. This property is used when the FP_PDFStructure property is True for the document, and the FrameMaker product generates PDF data. The value for this property can be between 0 and 100, where greater values are deeper in the hierarchy. If FP_PDFStructureLevel is 0, the FrameMaker product does not include paragraphs of this format in the PDF structure.

FMParagraphFormat Asian Character Spacing Properties

Property:	Type:	Description:
MaxJLetSpace	F_Metric	Maximum Asian letter space.
MaxJRomSpace	F_Metric	Maximum Asian-Roman space.
MinJLetSpace	F_Metric	Minimum Asian letter space.
MinJRomSpace	F_Metric	Minimum Asian-Roman space.
OptJLetSpace	F_Metric	Optimum Asian letter space.
OptJRomSpace	F_Metric	Optimum Asian-Roman space.
YakumonoType	F_Int	The Yakumono rules to handle punctuation characters; can be one of: mFV_FLOATING_YAKUMONO mFV_MONOSPACE_YAKUMONO mFV_FIXED_YAKUMONO

FMParagraphFormat Autonumbering Properties

Property:	Type:	Description:
AutoNumChar	F_String	Character Format for the automatic numbering string specified by utoNumString; "" if the default character format is used.
AutoNumString	F_String	Autonumber Format string; for example, "<n>.<n+>".
NumAtEnd	F_Int	True if numbering position is End of Paragraph; False if it is Beginning of Paragraph.
PgfIsAutoNum	F_Int	True if autonumbering is enabled.

FMParagraphFormat Default Font Properties

Property:	Type:	Description:
Capitalization	F_Int	Type of capitalization to use: mFV_CAPITAL_CASE_NORM mFV_CAPITAL_CASE_SMALL mFV_CAPITAL_CASE_LOWER mFV_CAPITAL_CASE_UPPER
ChangeBar	F_Int	True if Change Bars are on.
Color	FMColor	Spot color.
CombinedFontFamily	FMCombinedFont Definition	Combined font definition.
FontAngle	F_Int	Font angle (specifies an index into the array of font angles provided by the session property FontAngleNames).
FontEncodingName	F_String	The font's encoding.
FontFamily	F_Int	Font family (specifies an index into the array of font families provided by the session property FontFamilyNames).

Property:	Type:	Description:
FontPlatformName	F_String	Name that uniquely identifies a font on a specific platform. For more information, see “How FrameAC Represents Font Information” on page 417 of Appendix B, “FrameMaker Document and Session Architecture.”
FontPostScriptName	F_String	Name given to a font when it is sent to a PostScript printer. For more information, see “How FrameAC Represents Font Information” on page 417 of Appendix B, “FrameMaker Document and Session Architecture.”
FontSize	F_Metric	Font size (2 pt to 400 pt).
FontVariation	F_Int	Font variation (specifies an index into the array of font variations provided by the FMSession property FontVariationNames).
FontWeight	F_Int	Font weight (specifies an index into the array of font weights provided by the FMSession property FontWeightNames).
KernX	F_Metric	Horizontal kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves a character right and a negative value moves a character left.
KernY	F_Metric	Vertical kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves characters up and a negative value moves characters down.
Overline	F_Int	True if Overline is enabled.
PairKern	F_Int	True if Pair Kern is enabled.
Position	F_Int	Specifies position relative to baseline of text: mFV_POS_NORM: Normal mFV_POS_SUB: Subscript mFV_POS_SUPER: Superscript

Property:	Type:	Description:
Spread	F_Metric	Obsolete property, but still functional. See corresponding "tracking" property below.
Stretch	F_Metric	Character stretch (set width) expressed as a percentage of normal stretch for the font (metric –10% to 1000%).
Strikethrough	F_Int	True if Strikethrough is enabled.
Underlining	F_Int	Type of underlining: mFV_CB_NO_UNDERLINE mFV_CB_SINGLE_UNDERLINE mFV_CB_DOUBLE_UNDERLINE mFV_CB_NUMERIC_UNDERLINE

FMParagraphFormat Table Cell Properties

Property:	Type:	Description:
CellBottomMargin	F_Metric	Amount added to default bottom margin of table cell.
CellLeftMargin	F_Metric	Amount added to default left margin of table cell.

Property:	Type:	Description:
CellMarginsFixed	F_Int	Specifies which cell margins are fixed. The following values can be ORed into it: mFV_PGF_FIXED_B_MARGIN: the bottom margin is fixed mFV_PGF_FIXED_L_MARGIN: the left margin is fixed mFV_PGF_FIXED_R_MARGIN: the right margin is fixed mFV_PGF_FIXED_T_MARGIN: the top margin is fixed If the margin for a cell is fixed, the margin property specifies the absolute value of the cell margin. For example, if mFV_PGF_FIXED_B_MARGIN is set, CellBottomMargin specifies the absolute value of the cell's bottom margin, overriding the cell margin specified by the table format. If mFV_PGF_FIXED_B_MARGIN is not set, CellBottomMargin is added to the margin specified by the table format.
CellRightMargin	F_Metric	Amount added to default right margin of table cell.
CellTopMargin	F_Metric	Amount added to default top margin of table cell.
CellVAlignment	F_Int	Vertical alignment of a paragraph when it is the first one in a cell: mFV_PGF_V_ALIGN_TOP mFV_PGF_V_ALIGN_MIDDLE mFV_PGF_V_ALIGN_BOTTOM

FMParagraphFormat Hyphenation Properties

Property:	Type:	Description:
AdjHyphens	F_Int	Number of allowable adjacent hyphens.
Hyphenate	F_Int	True if Automatic Hyphenation is enabled.

Property:	Type:	Description:
HyphMinPrefix	F_Int	Minimum number of letters that must precede hyphen.
HyphMinSuffix	F_Int	Minimum number of letters that must follow a hyphen.
HyphMinWord	F_Int	Minimum length of a hyphenated word.

FMParagraphFormat Linguistic Properties

The following properties describe the paragraph's language.

Property:	Type:	Description:
Language	F_Int	Hyphenation and spell-checking language to use: mFV_LANG_BRAZILIAN mFV_LANG_BRITISH mFV_LANG_CANADIAN_FRENCH mFV_LANG_CATALAN mFV_LANG_DANISH mFV_LANG_DUTCH mFV_LANG_ENGLISH mFV_LANG_FINNISH mFV_LANG_FRENCH mFV_LANG_GERMAN mFV_LANG_ITALIAN mFV_LANG_NOLANGUAGE mFV_LANG_NORWEGIAN mFV_LANG_NYNORSK mFV_LANG_PORTUGUESE mFV_LANG_SPANISH mFV_LANG_SWEDISH mFV_LANG_SWISS_GERMAN mFV_LANG_JAPANESE mFV_LANG_TRADITIONAL_CHINESE mFV_LANG_SIMPLIFIED_CHINESE mFV_LANG_KOREAN

FMParagraphFormat Indent Properties

Property:	Type:	Description:
FirstIndent	F_Metric	First-line left margin, measured from left side of current text column (0 cm to 100 cm).

Property:	Type:	Description:
LeftIndent	F_Metric	Left margin, measured from left side of current text column (0 cm to 100 cm).
RightIndent	F_Metric	Right margin, measured from right side of current text column.

FMParagraphFormat Line Spacing Properties

Property:	Type:	Description:
Leading	F_Metric	Space below each line in a paragraph.
LineSpacing	F_Int	Space between lines in a paragraph measured from baseline to baseline: mFV_PGF_FIXED: default font size mFV_PGF_PROPORTIONAL: largest font in line mFV_PGF_FLOATING: largest ascender in line

FMParagraphFormat Placement Properties

Property:	Type:	Description:
BlockLines	F_Int	The number of Widow/Orphan lines.
KeepWithNext	F_Int	True if Keep With Next Paragraph is enabled.
KeepWithPrev	F_Int	True if Keep With Previous Paragraph is enabled.
PgfAlignment	F_Int	Horizontal alignment of paragraph: mFV_PGF_LEFT mFV_PGF_RIGHT mFV_PGF_CENTER mFV_PGF_JUSTIFIED

Property:	Type:	Description:
Placement	F_Int	Paragraph placement: mFV_PGF_SIDEBODY mFV_PGF_SIDEHEAD_TOP mFV_PGF_SIDEHEAD_FIRST_BASELIN E mFV_PGF_SIDEHEAD_LAST_BASELINE mFV_PGF_RUN_IN mFV_PGF_STRADDLE mFV_PGF_STRADDLE_NORMAL_ONLY
RunInSeparator	F_String	String for Run-In Head Default Punctuation.
SpaceAbove	F_Metric	Space above paragraph.
SpaceBelow	F_Metric	Space below paragraph.
Start	F_Int	Vertical placement of paragraph: mFV_PGF_ANYWHERE mFV_PGF_TOP_OF_COL mFV_PGF_TOP_OF_PAGE mFV_PGF_TOP_OF_LEFT_PAGE mFV_PGF_TOP_OF_RIGHT_PAGE

FMParagraphFormat Object Pointer Properties

Property:	Type:	Description:
NextPgfFmtInDoc	F_Int (Obj ID)	Next FMParagraph in the document.

FMParagraphFormat Reference Frame Properties

Property:	Type:	Description:
BottomSeparator	F_String	Name of frame to put below paragraph.
TopSeparator	F_String	Name of frame to put above paragraph.

FMParagraphFormat Tab Properties

The following properties specify the tab settings in the paragraph (not tab characters).

Property:	Type:	Description:
NumTabs	F_Int	Number of tabs in the paragraph.
Tabs	Array of udtTab	Array of tab descriptions that specify the positions and types of tab stops.

FMParagraphFormat Tagging Properties

Property:	Type:	Description:
Name	F_String	Name of paragraph format.
NextTag	F_String	Tag for new next paragraph.
UseNextTag	F_Int	True if Next Paragraph Tag is enabled.

FMParagraphFormat Word Spacing Properties

Property:	Type:	Description:
LetterSpace	F_Int	True if Word Spacing is enabled.
MaxSpace	F_Metric	Maximum word spacing (percentage of an em space in current font).
MinSpace	F_Metric	Minimum word spacing (percentage of an em space in current font).
OptSpace	F_Metric	Optimum word spacing.

FMPolyLine

Details

FMPolyLine objects are drawn objects that describe straight-line segments joined at multiple vertices.

See also

"FMObject" on page 166, "FMGraphic" on page 155.

Properties

Property:	Type:	Description:
NumPoints	F_Int	Number of the polyline's vertices
Points	Array of udtPoint	Array of x-y coordinate pairs that specify the polyline's vertices
PolyIsBezier	F_Int	True if polyline is a Bezier curve

FMPolygon

Details

FMPolyLine objects are drawn objects that describe a closed figure of straight-line segments with multiple vertices.

See also

"FMObject" on page 166, "FMGraphic" on page 155.

Properties

Property:	Type:	Description:
NumPoints	F_Int	Number of the polygon's vertices
Points	Array of udtPoint	Array of x-y coordinate pairs that specify the polygon's vertices
PolyIsBezier	F_Int	True if polygon is smoothed

FMRectangle

Details

FMRectangle objects represent drawn rectangles.

See also

"FMObject" on page 166, "FMGraphic" on page 155.

Properties

Property:	Type:	Description:
RectangleIsSmoothed	F_Int	True if smoothing is enabled

FMReferencePage

Details

FMReferencePage objects represent the reference pages in a document.

See also

"FMObject" on page 166, "FMPage" on page 168.

Properties

Property:	Type:	Description:
Name	F_String	Name of reference page
PageNum	F_Int	Page number

FMRoundRectangle

Details

FMRoundRectangle objects represent drawn rectangles with rounded corners.

See also

"FMObject" on page 166, "FMGraphic" on page 155.

Properties

Property:	Type:	Description:
Radius	F_Metric	Radius of corner; 0 for a square corner

FMRow

Details

FMRow objects represent rows in FMTable objects. One row contains a number of FMCell objects—the number of cells in a row indicates the number of columns in the table.

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
CondFmtIsShown	F_Int	True if the condition is shown.
Element	FMElement	If the row is in a FrameMaker+SGML document, the ID of the element containing the row.
FirstCellInRow	FMCell	First FMCell in row.
Height	F_Metric	Height of the row.
InCond	Array of FMConditionalFormat objects	Condition tags for row.
LocX	F_Metric	Offset from the left side of the FMTextFrame containing the row.
LocY	F_Metric	Offset from the top of the FMPageFrame containing the row.
NextRowInTbl	FMRow	Next FMRow in the table.
PrevRowInTbl	FMRow	Previous FMRow in the table.
RowIsShown	F_Int	True if the conditional row is shown.
RowKeepWithNext	F_Int	True if Keep With Next Row is enabled.
RowKeepWithPrev	F_Int	True if Keep With Previous Row is enabled.
RowMaxHeight	F_Metric	Maximum row height.
RowMinHeight	F_Metric	Minimum row height.
RowStart	F_Int	Row placement: mFV_ROW_ANYWHERE mFV_ROW_TOP_OF_COL mFV_ROW_TOP_OF_PAGE mFV_ROW_TOP_OF_LEFT_PAGE mFV_ROW_TOP_OF_RIGHT_PAGE
RowTbl	FMTable	FMTable containing the row.

Property:	Type:	Description:
RowType	F_Int	Type of row: mFV_ROW_HEADING mFV_ROW_BODY mFV_ROW_FOOTING
SepOverride	FMColor	Color separation format override.
StyleOverrides	F_Int	Style condition indicators for conditional text: mFV_CS_DOUBLE_UNDERLINE mFV_CS_NO_OVERRIDE mFV_CS_OVERLINE mFV_CS_SINGLE_UNDERLINE mFV_CS_STRIKETHROUGH All style condition indicators are represented as hatched lines for the table rows.
UseSepOverride	F_Int	True if SepOverride property overrides default from the table.
Width	F_Metric	Width of the row.
Cells	Array of FMCell	The cells in the current row.

Methods:

Method:	Description:
IsRowEmpty	Returns True if all the cells in the row are empty.
DeleteRowText	Deletes the contents of the row.

FMRubi

Details

FMRubi objects represent rubi composites, which are used for Japanese text in a document. A rubi composite is typically a word written in Kanji and Hiragana, with the pronunciation spelled out over the word in smaller Hiragana characters.

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
Element	FMElement	If the rubi group is in a structured document, the object handle of the associated FMElement for the rubi group element.
NextRubiInDoc	FMRubi	The next instance of a rubi composite in the document..
OyamojiTextRange	udtTextRange	The text range the oyamoji text encompasses.
RubiElement	FMElement	If the rubi group is in a structured document, the object handle of the associated FMElement for the rubi element.
RubiTextRange	udtTextRange	The text range the rubi text encompasses.
Unique	F_Int	The rubi composite's UID.

FMRulingFormat

Details

FMRulingFormat objects represent the table ruling format definitions that are stored in a document. You apply FMRulingFormat objects to a table or a table format. See "Table Ruling Properties" on page 207 and "Table Format Ruling Properties" on page 213 for more information.

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
Name	F_String	Ruling format name.
NextRulingFmtInDoc	FMRulingFormat	Next ruling format in document.
Pen	F_Int	The pen pattern (numbers between 0 and 7; see "Pen and FIll properties" on page 407 in Appendix A, "Special Cases for Object Properties." The FDK provides constants for several pen patterns: mFV_FILL_BLACK mFV_FILL_WHITE mFV_FILL_CLEAR
RulingGap	F_Metric	Gap between double ruling lines (0.015 pt to 360 pt).
RulingLines	F_Int	Number of ruling lines (0 to 2 lines).
RulingPenWidth	F_Metric	Ruling line thickness (0.015 pt to 360 pt).
RulingSep	FMColor	Spot color of ruling format.

FMSession

Details

The FMSession object is the entry into a FrameMaker session. From there you can use its methods to create documents or initialize your programs, and you can use properties to get session information such as the currently active book or document, a list of open documents or books, a list of registered filters, or a list of available fonts.

Properties

Property:	Type:	Description:
ActiveBook	FMBook	The book with input focus.
ActiveDoc	FMDocument	The document with input focus.
AddMarkerTypeToStandardMarkers	F_String	The name of a marker type to add to the standard list of marker types. Use F_ApiSetString() to set a marker type name to this property of the mFV_SessionId. See "Standard Marker Types for a Document" on page 419 in Appendix B, "FrameMaker Document and Session Architecture."
ApplyFormatRules	F_Int	True if element reformatting is enabled (FrameMaker+SGML only).
AutoBackup	F_Int	True if Automatic Backup is enabled.
AutoSave	F_Int	True if Automatic Save is enabled.
AutoSaveSeconds	F_Int	Time between automatic saves in seconds (60 seconds to 10800 seconds).
CurrentMenuSet	F_Int	Type of menu set: mFV_MENU_QUICK mFV_MENU_COMPLETE mFV_MENU_CUSTOM
DefaultFontAngle	F_Int	Index of the default font angle in the current session.
DefaultFontFamily	F_Int	Index of the default font family name in the current session.
DefaultFontVariation	F_Int	Index of the default font variation in the current session.
DefaultFontWeight	F_Int	Index of the default font weight in the current session.
Displaying	F_Int	False if screen refresh is completely turned off.
ExportFilters	Array of F_String	List of export filters available in the current session.

Property:	Type:	Description:
FirstCommandInSession	FMCommand	First command in the list of commands in the session (FMCommand).
FirstMenuItemInSession	FMMenu	First menu item or menu in the list of menus, menu items, and menu item separators in the session (FMCommand, FMMenu, FMMenuItemSeperator).
FirstOpenBook	FMBook	First open book in session.
FirstOpenDoc	FMDocument	First open document in session.
FM_BinDir	F_String	Directory pathname of $FMHOME/bin.
FM_CurrentDir	F_String	Name of the directory from which the FrameMaker product was started.
FM_HelpDir	F_String	Pathname of the FrameMaker product help directory
FM_HomeDir	F_String	Pathname of $FMHOME.
FM_InitDir	F_String	Directory pathname of $FMHOME/fminit.
FM_SgmlDir	F_String	Directory pathname of $FMHOME/structure/sgml.
FM_StructureDir	F_String	Directory pathname of $FMHOME/structure.
FM_XmlDir	F_String	Directory pathname of $FMHOME/structure/xml.
FontAngleNames	Array of F_String	List of font angles available in the current session.

Property:	Type:	Description:
FontFamilyAttributes	Array of F_Int	An array of flags that indicate attributes for each font family listed by FontFamilyNames. This array of integers is indexed the same as the list of font family names, and corresponds directly to that list. Each F_Int is a packed field; the high order 16 bits indicate a surrogate font, and the low order bits indicate attributes for the font family. The flags, their mask values, and their meaning follow: mFV_FAMILY_VISIBLE (0x00000001): Family is visible in menu. mFV_FAMILY_SELECTABLE (0x00000002): Family can be selected in menu. mFV_FAMILY_MAPPED (0x00000004): Family is is always mapped to another family. mFV_FAMILY_SURROGATE (0xFFFF0000): The family mapped to if mFV_FAMILY_MAPPED is True.
FontFamilyNames	Array of F_String	List of font family names available in the current session. Note that this list does not include FMCombinedFontDefinition objects.
FontVariationNames	Array of F_String	List of font variations available in the current session.
FontWeightNames	Array of F_String	List of font weights available in the current session.
Gravity	F_Int	True if Gravity is turned on for the session.
GreekSize	F_Metric	Size at which to greek text.
HostName	F_String	Name of the host computer.

Property:	Type:	Description:
IconBarOn	F_Int	True if the four icons that appear on the upper-right side of the document window are on. Changing this property affects only documents that are opened subsequently; it does not affect documents that are already open.
ImportFilters	Array of F_String	List of import filters available in the current session.
IsIconified	F_Int	True if the FrameMaker product window is iconified.
IsInFront	F_Int	True if the FrameMaker product window is in front of other application windows. You can use this property to bring the FrameMaker product to the front or back.
Label	F_String	The title in the FrameMaker product window title bar.
Language	F_Int	Product language: mFV_LANG_BRITISH mFV_LANG_ENGLISH mFV_LANG_FRENCH mFV_LANG_GERMAN mFV_LANG_ITALIAN mFV_LANG_NOLANGUAGE mFV_LANG_SPANISH mFV_LANG_SWEDISH mFV_LANG_JAPANESE mFV_LANG_TRADITIONAL_CHINESE mFV_LANG_SIMPLIFIED_CHINESE mFV_LANG_KOREAN

Property:	Type:	Description:
MarkerNames	Array of F_String	List of standard marker types for the current session. For versions prior to 5.5, this returned the list of all marker types for the current session. In version 5.5, marker types are assigned to the document; use the MarkerTypeNames property of FMDocument to get the full list of marker types.
OpenDir	F_String	The FrameMaker product directory.
OperatingSystem	F_String	Operating system under which the current session is running: Solaris HPUX AIX Macintosh DOS
Path	F_String	Pathname to search to start the FrameMaker product.
Platform	F_String	Name of the platform on which the current session is running: Sun HP RS6000 Macintosh Intel
ProductIsDemo	F_Int	True if the current session is for a demo version of FrameMaker
ProductIsStructured	F_Int	True if FrameMaker is running in structured mode for the current session

Property:	Type:	Description:
ProductName	F_String	The FrameMaker product name. The names for FrameMaker+SGML indicate FrameMaker running under the structured product interface. FrameViewer is retained for backward compatibility. Can be one of: FrameMaker FrameMaker+SGML+SGML FrameViewer DemoMaker DemoMaker+SGML
Reformatting	F_Int	True if reformatting is enabled.
RememberMissingFontNames	F_Int	True if Remember Missing Font Names is activated .
ScreenHeight	F_Int	Height of the FrameMaker product window in pixels.
ScreenWidth	F_Int	Width of the FrameMaker product window in pixels.
ScreenX	F_Int	The offset of the FrameMaker product window in pixels from the left side of the screen. If you set a value that would result in the product window being off the screen, that value is ignored and the old value is retained.
ScreenY	F_Int	The offset of the FrameMaker product window in pixels from the top of the screen. If you set a value that would result in the product window being off the screen, that value is ignored and the old value is retained.
Snap	F_Int	True if Snap is turned on for the session.

Property:	Type:	Description:
TmpDir	F_String	Pathname of temporary directory for internal FrameMaker product processes; the directory specified by the $TEMP environment variable (on Windows).
UserLogin	F_String	User login name.
UserName	F_String	User name.
Validating	F_Int	True if validation is enabled (FrameMaker+SGML only).
VersionMajor	F_Int	Frame version number (before the decimal).
VersionMinor	F_Int	Frame version number (after the decimal).
ViewFormattingBar	F_Int	True if the formatting bar is visible.
ViewQuickAccessBar	F_Int	True if the QuickAccess bar is visible.
WindowSystem	F_String	Name of window system that the FrameMaker product is running under: Macintosh MSWindows X Windows
Documents	Array of FMDocument	List of open documents.
Books	Array of FMBook	List of open books.
FA_Errno	F_Int	An integer that regiesters the error state after calling a FrameAC method. To use this integer, you should set it to mFE_Success before executing the method, then check the value immediately afterward. This can be an aid to debugging your programs.

Methods:

Method:	Description:
Alert	Displays an alert box.

Methods: (Continued)

Method:	Description:
BailOut	Sets up your FrameAC process to free its memory when it finishes its processing.
CallClient	Calls a registered FDK client. Calls FrameAC as a single FDK client, then passes the call to all FrameAC applications that requested the notification.
CustomDoc	Creates an untitled document of the specified size.
DefineCommand	Creates a menu command.
DefineMenu	Creates a menu.
OpenDocument	Opens a document using a list of properties as a script.
OpenBook	Opens a book using a list of properties as a script.
NewDocument	Creates an untitled document using a list of properties as a script.
NewBook	Creates an untitled book using a list of properties as a script.
SimpleOpenDocument	Unscripted document open.
SimpleOpenBook	Unscripted book open.
GetNamedObject	Gets a named session object.
GetObject	Gets a session object.
GetOpenDefaultParams	Gets the default script for opening a document or book.
GetSaveDefaultParams	Gets the default script for saving a document or book.
SetNotification	Sets a notification point for the OnApiNotify event.
Initialise	Initialises a FrameAC plug-in.
GetImportDefaultParams	Gets the default script for importing a file.
ExecuteScript	Executes a FrameAC script.

FMSubCol

Details

FMSubCol objects represent the columns that are defined within a FMTextFrame object. The text frame may have one or more FMSubCol objects. FMSubCol objects do not inherit any properties or methods from FMGraphic. To get the text from a sub column, use the GetText method that the sub column inherits from FMObject.

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
ContentHeight	F_Metric	The distance between the top of the column and the baseline of the last line in the column.
FirstAFrame	FMAnchoredFrame	First anchored frame in the column.
FirstCell	FMCell	First table cell in the column.
FirstFn	FMFootnote	First footnote in the colum.
FirstPgf	FMParagraph	First paragraph in the column.
FrameParent	FMTextFrame	ID of text frame that contains the column.
Height	F_Metric	Column height.
LastAFrame	FMAnchoredFrame	Last anchored frame in the column.
LastCell	FMCell	Last table cell in the column.
LastFn	FMFootnote	Last footnote in the column.
LastPgf	FMParagraph	Last paragraph in the column.
LocX	F_Metric	Offset from left side of the text frame that contains the column.
LocY	F_Metric	Offset from top of text frame that contains the column.
NextSubCol	FMSubCol	Next column in the flow.

Property:	Type:	Description:
Overflowed	F_Int	True if the text frame containing the column has Autoconnect turned off and text overflows the column.
ParentTextFrame	FMTextFrame	ID of text frame that contains the column.
PrevSubCol	FMSubCol	Previous column in the flow.
Unique	F_Int	Text column's UID.
Width	F_Metric	Column width.

FMTable

Details

FMTable objects represent tables in a document. To exist in a document, a table must have at least one body FMRow object.

See also

"FMObject" on page 166.

Properties

This reference manual divides the FMTable properties into the following groups:

- "Table Basic Properties" on page 205
- "Table General Properties" on page 207
- "Table Ruling Properties" on page 207
- "Table Selection Properties" on page 208
- "Table Shading and Colour Properties" on page 209
- "Table Structure Properties" on page 209
- "Table Title Properties" on page 210

Table Basic Properties

Property:	Type:	Description:
ContentHeight	F_Metric	The height of the table title.

Property:	Type:	Description:
Locked	F_Int	True if the table is part of a text inset that retains formatting information from the source document. The table is not affected by global formatting performed on the document.
OrphanRows	F_Int	Number of orphan rows.
Overflowed	F_Int	True if the table has cells that are not shown because they extend beyond the text frame boundaries.
TblAlignment	F_Int	Horizontal placement of table: mFV_ALIGN_TBL_CENTER mFV_ALIGN_TBL_LEFT mFV_ALIGN_TBL_RIGHT
TblCellBottomMargin	F_Metric	Default bottom cell margin for the table.
TblCellLeftMargin	F_Metric	Default left cell margin for the table.
TblCellRightMargin	F_Metric	Default right cell margin for the table.
TblCellTopMargin	F_Metric	Default top cell margin for the table.
TblInLockedTi	F_Int	True if the table is in a locked text inset. You should not use FrameAC to delete table objects when the table is in a locked text inset.
TblLeftIndent	F_Metric	Left indent for the table.
TblPlacement	F_Int	Vertical placement of table on page: mFV_TBL_ANYWHERE mFV_TBL_TOP_OF_COL mFV_TBL_TOP_OF_PAGE mFV_TBL_TOP_OF_LEFT_PAGE mFV_TBL_TOP_OF_RIGHT_PAGE mFV_TBL_FLOAT
TblRightIndent	F_Metric	Right indent for the table.
TblSpaceAbove	F_Metric	Vertical space above the table.
TblSpaceBelow	F_Metric	Vertical space below the table.

Property:	Type:	Description:
TextLoc	udtTextLoc	The text location of the table's anchor.
Unique	F_Int	The table's UID.

Table General Properties

Property:	Type:	Description:
FirstRowInTbl	FMRow	First row in the table.
LastRowInTbl	FMRow	Last row in the table.
NextTblInDoc	FMTable	Next table in the document.
Rows	Array of FMRow	The rows in the table.
TblCatalogEntry	F_Int	True if the table's format is in the Table Catalog.
TblColWidths	Array of F_Metric	List of column widths.
TblNumbering	F_Int	Direction of autonumbering for the table: mFV_TBL_NUM_BY_COL mFV_TBL_NUM_BY_ROW
TblNumCols	F_Int	Number of columns in the table.
TblNumRows	F_Int	Number of rows in the table.
TblTag	F_String	Name of table format.
TblWidth	F_Metric	Horizontal width of the table.

Table Ruling Properties

Property:	Type:	Description:
TblBodyRowRuling	FMRulingFormat	Ruling applied to body rows specified by TblBodyRowRulingPeriod.
TblBodyRowRulingPeriod	F_Int	The periodicity of the ruling specified by TblBodyRowRuling. For example, if TblBodyRowRulingPeriod is set to 3, the ruling specified by TblBodyRowRuling is applied to every third row.

Property:	Type:	Description:
TblBottomRuling	FMRulingFormat	Ruling applied to the bottom of the table.
TblColRuling	FMRulingFormat	Ruling applied to table columns specified by TblColRulingPeriod.
TblColRulingPeriod	F_Int	The periodicity of the ruling specified by TblColRuling. For example, if TblColRulingPeriod is set to 2, the ruling specified by TblColRuling is applied to every other column.
TblHFRowRuling	FMRulingFormat	Ruling for table heading and footing rows.
TblHFSeparatorRuling	FMRulingFormat	Separator ruling for table heading and footing rows.
TblLastBodyRuling	FMRulingFormat	True if Draw Bottom Ruling on Last Sheet Only is enable.
TblLeftRuling	FMRulingFormat	Ruling for the left side of the table.
TblOtherBodyRowRuling	FMRulingFormat	Ruling for body rows that aren't specified by FP_TblBodyRowRulingPeriod.
TblOtherColRuling	FMRulingFormat	Ruling for columns that aren't specified by FP_TblColRulingPeriod.
TblRightRuling	FMRulingFormat	Ruling for the right side of the table.
TblTopRuling	FMRulingFormat	Ruling for the top of the table.

Table Selection Properties

FMTable objects have the following properties that specify a table's selected rows and columns. All table selection properties are read-only. To select table rows or cells, use the MakeTblSelection method.

Property:	Type:	Description:
BottomRowSelection	FMRow	Bottom body row in selection, if table is selected.
LeftColNum	F_Int	Number of leftmost selected column, if table is selected (columns are numbered from left to right, starting with 0)

Property:	Type:	Description:
RightColNum	F_Int	Number of rightmost selected column, if table is selected (columns are numbered from left to right, starting with 0).
TblTitleSelected	F_Int	True if table title is selected.
TopRowSelection	FMRow	Top row in selection, if table is selected.

Table Shading and Colour Properties

Property:	Type:	Description:
TblBodyFirstColor	FMColor	First spot color for table body.
TblBodyFirstFill	F_Int	First fill pattern for table body.
TblBodyFirstPeriod	F_Int	Number of columns or body rows to which the first fill pattern (specified by TblBodyFirstFill) is applied.
TblBodyNextColor	FMColor	Exception color for columns or body rows.
TblBodyNextFill	F_Int	Exception fill pattern for table body
TblBodyShadeBy	F_Int	True if Shade By is set to Columns; False if Shade By is set to Rows.
TblHFColor	FMColor	Color for table heading and footing.
TblHFFill	F_Int	Fill pattern for table heading and footing (integer percentage).

Table Structure Properties

Property:	Type:	Description:
Element	FMElement	The element associated with the table.
TblBodyElement	FMElement	The element containing the table's body rows.
TblElement	FMElement	The element containing the table.
TblFooterElement	FMElement	The element containing the table's footer rows.

Property:	Type:	Description:
TblHeaderElement	FMElement	The element containing the table's header rows.
TblTitleElement	FMElement	The element containing the table title.

Table Title Properties

Property:	Type:	Description:
FirstPgf	FMParagraph	If table has a title, the first paragraph in the title.
HighestLevelElement	FMElement	If table is in a structured document and has a title, the title's highest-level element. HighestLevelElement is obsolete but is supported for backward compatibility.
LastPgf	FMParagraph	If table has a title, the last paragraph in the title.
TblTitleGap	F_Metric	Gap between the title and top or bottom row of the table.
TblTitlePosition	F_Int	The placement of the table title: mFV_TBL_NO_TITLE: table has no title mFV_TBL_TITLE_BELOW: the title appears below the table mFV_TBL_TITLE_ABOVE: the title appears above the table
TblTitleSelected	F_Int	True if table title is selected

Methods:

Method:	Description:
AddCols	Adds one or more columns to the table.
AddRows	Adds one or more rows to the table—the rows can be heading, footing, or body rows.
DeleteCols	Deletes one or more columns.
DeleteRows	Deletes one or more rows.

Methods: (Continued)

Method:	Description:
MakeTblSelection	Selects cells in a table.
GetCountOfRowsOfType	Gets the number of heading, footing, or body rows in the table.
GetFirstRow	Gets the first row in the table.
GetLastRow	Gets the last row in the table.
DeleteBodyRows	Deletes the body rows from a table, leaving one empty body row.

FMTableFormat

Details

FMTableFormat objects represent the table formats that are stored in a document's table format catalog.

See also

"FMObject" on page 166.

Properties

This reference manual divides the FMTableFormat properties into the following groups:

- "Table Format Basic Properties" on page 211
- "Table Format General Properties" on page 212
- "Table Format New Table Properties" on page 213
- "Table Format Ruling Properties" on page 213
- "Table Format Shading and Colour Properties" on page 214

Table Format Basic Properties

FMTableFormat objects have the following properties that specify a table's indents, alignment, and other placement characteristics. Note that a FMTableFormat object also includes all the properties for a FMParagraphFormat object in order to describe the properties of the table title.

Property:	Type:	Description:
OrphanRows	F_Int	Number of orphan rows

Property:	Type:	Description:
TblAlignment	F_Int	Horizontal placement of table: mFV_ALIGN_TBL_CENTER mFV_ALIGN_TBL_LEFT mFV_ALIGN_TBL_RIGHT
TblCellBottomMargin	F_Metric	Default bottom cell margin for the table.
TblCellLeftMargin	F_Metric	Default left cell margin for the table.
TblCellRightMargin	F_Metric	Default right cell margin for the table.
TblCellTopMargin	F_Metric	Default top cell margin for the table.
TblLeftIndent	F_Metric	Left indent for the table.
TblPlacement	F_Int	Vertical placement of table on page: mFV_TBL_ANYWHERE mFV_TBL_TOP_OF_COL mFV_TBL_TOP_OF_PAGE mFV_TBL_TOP_OF_LEFT_PAGE mFV_TBL_TOP_OF_RIGHT_PAGE mFV_TBL_FLOAT
TblRightIndent	F_Metric	Right indent for table.
TblSpaceAbove	F_Metric	Vertical space above table.
TblSpaceBelow	F_Metric	Vertical space below table.
TblTitleGap	F_Metric	Gap between title and top or bottom row.
TblTitlePosition	F_Int	Placement of table title: mFV_TBL_NO_TITLE mFV_TBL_TITLE_BELOW mFV_TBL_TITLE_ABOVE

Table Format General Properties

The FMTableFormat object inherits the Name property from the FMParagraphFormat object. This Name property is the name of the paragraph format that is applied to the first paragraph in the table title. To get the name of the table format, use the TblTag property.

Property:	Type:	Description:
TblCatalogEntry	F_Int	True if format is in the Table Catalog.

Property:	Type:	Description:
NextTblFmtInDoc	FMTableFormat	Next table format in the document.
TblNumbering	F_Int	Direction of autonumbering for the table: mFV_TBL_NUM_BY_COL mFV_TBL_NUM_BY_ROW
TblTag	F_String	Name of the table format.

Table Format New Table Properties

Property:	Type:	Description:
TblInitNumBodyRows	F_Int	Number of body rows for new table.
TblInitNumCols	F_Int	Number of columns for new table.
TblInitNumFRows	F_Int	Number of footing rows for new table.
TblInitNumHRows	F_Int	Number of heading rows for new table.

Table Format Ruling Properties

Property:	Type:	Description:
TblBodyRowRuling	FMRulingFormat	Ruling for body rows that aren't specified by TblBodyRowRulingPeriod.
TblBodyRowRulingPeriod	F_Int	The periodicity of the ruling specified by TblOtherBodyRowRuling. For example, if TblBodyRowRulingPeriod is set to 3, the ruling specified by TblOtherBodyRowRuling is applied to every third row.
TblBottomRuling	FMRulingFormat	Ruling for the bottom of the table.
TblColRuling	FMRulingFormat	Ruling for columns that aren't specified by FP_TblColRulingPeriod.

Property:	Type:	Description:
TblColRulingPeriod	F_Int	The periodicity of the ruling specified by TblOtherColRuling. For example, if TblColRulingPeriod is set to 2, the ruling specified by TblOtherColRuling is applied to every other column.
TblHFRowRuling	FMRulingFormat	Ruling for the heading and footing rows (FMRulingFormat).
TblHFSeparatorRuling	FMRulingFormat	Separator ruling for the table heading and footing rows (FMRulingFormat).
TblLastBodyRuling	FMRulingFormat	True if Draw Bottom Ruling on Last Sheet Only is enabled.
TblLeftRuling	FMRulingFormat	Ruling for the left side of the table.
TblOtherBodyRowRuling	FMRulingFormat	Ruling applied to body rows specified by FP_TblBodyRowRulingPeriod.
TblOtherColRuling	FMRulingFormat	Ruling applied to table columns specified by FP_TblColRulingPeriod.
TblRightRuling	FMRulingFormat	Ruling for the right side of the table.
TblTopRuling	FMRulingFormat	Ruling for the top of the table.

Table Format Shading and Colour Properties

Property:	Type:	Description:
TblBodyFirstColor	FMColor	First spot color for table body.
TblBodyFirstFill	F_Int	First fill pattern for table body.
TblBodyFirstPeriod	F_Int	Number of columns or body rows to which the first fill pattern (specified by TblBodyFirstFill) is applied.
TblBodyNextColor	FMColor	Exception color for columns or body rows.
TblBodyNextFill	F_Int	Exception fill pattern for table body.
TblBodyNextPeriod	F_Int	Number of columns or body rows to which the exception fill pattern (specified by TblBodyNextFill) is applied.

Property:	Type:	Description:
TblBodyShadeBy	F_Int	True if Shade By is set to Columns; False if Shade By is set to Rows.
TblHFColor	FMColor	Color for table heading and footing.
TblHFFill	F_Int	Fill pattern for table heading and footing (integer percentage).

FMTextFrame

Details

FMTextFrame objects represent the text frames in a document. The text frame may have one or more FMSubCol objects. To get the text from a text frame, use the GetText method that the text frame inherits from FMObject.

See also

"FMObject" on page 166, "FMGraphic" on page 155.

Properties

Property:	Type:	Description:
ColGapWidth	F_Metric	Gap between columns (0 to 50 inches).
ColumnsAreBalanced	F_Int	True if terminal and underfilled columns in the flow are balanced.
FirstAFrame	FMAnchoredFrame	First anchored frame in the text frame.
FirstCell	FMCell	First table cell in the text frame.
FirstFn	FMFootnote	First footnote in the text frame.
FirstPgf	FMParagraph	First paragraph in the text frame.
FirstSubCol	FMSubCol	First column in the text frame.
Flow	FMFlow	Flow containing the text frame.
GraphicIsButton	F_Int	True if the text frame is a hypertext button.
LastAFrame	FMAnchoredFrame	Last anchored frame in the text frame.
LastCell	FMCell	Last table cell in the text frame.
LastFn	FMFootnote	Last footnote in the text frame.
LastPgf	FMParagraph	Last paragraph in the text frame.
LastSubCol	FMSubCol	Last column in the text frame.
NextTextFrameInFlow	FMTextFrame	Next text frame in the flow.
NumColumns	F_Int	The number of columns in the underlying column grid (1–10).
PrevTextFrameInFlow	FMTextFrame	Previous text frame in the flow.
SideHeadGap	F_Metric	Gap between side head area and body text area (0 to 50 inches).
SideHeadPlacement	F_Int	Placement of side heads relative to columns in the text frame: mFV_SH_LEFT mFV_SH_RIGHT mFV_SH_INSIDE mFV_SH_OUTSIDE

Property:	Type:	Description:
SideHeadWidth	F_Metric	Width of side head area for the text frame (0 to 50 inches).
Text	F_String	The text content of the text inset.

FMTextInset

Details

FrameAC uses four different types of text insets—FMTextInset_ApiClient, FMTextInset_Flow, FMTextInset_Text, and FMTextInset_TextTable. The FMTextInset object is the ancestor of those objects, and provides them with a common set of properties and methods.

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
ImportHint	F_String	Record identifying the filter used to import the text. The FrameMaker product uses this record to find the filter to use when updating the inset. For a complete description of the syntax of this string, see "Graphic Inset Import Hint Strings" on page 407 in Appendix A, "Special Cases for Object Properties."
LastUpdate	F_Int	Time when the inset was last updated, expressed in seconds since 1 January, 1970.
Name	F_String	A name assigned to the inset by a FrameAC plug-in or FDK client. It is not automatically assigned by the FrameMaker product.

Property:	Type:	Description:
NextTiInDoc	FMTextInset	The ID of the next text inset in the list of text insets in the document (FMTextInset_ApiClient, FMTextInset_Flow, FMTextInset_Text, or FMTextInset_TextTable).
TextRange	udtTextRange	The text range, in the document containing the text inset, occupied by the text inset.
TiAutomaticUpdate	F_Int	True if the inset is updated automatically. TiAutomaticUpdate has no effect if the document's DontUpdateTextInsets property is set to True.
TiFile	F_String	Pathname of the source file.
TiFileModDate	F_String	The modification date of the text inste's source file.
TiLocked	F_Int	True if the inset is locked. To change an inset's contents, you must unlock it. Always relock an inset after you have finished changing its contents.
TiMacEdition	F_Int	If the source file is a Macintosh edition, the ID of its sect and alis records.
Unique	F_Int	The text inset's UID.

Methods:

Method:	Description:
DeleteTextInsetContents	Deletes the content of the text inset.
UpdateTextInset	Updates the text inset.

FMTextInset_ApiClient

Details

FMTextInset_ApiClient objects represent text insets that are controlled by FrameAC programs or other FrameMaker plugins.

See also

"FMObject" on page 166, "FMTextInset" on page 217.

Properties

Property:	Type:	Description:
TiClientData	F_String	Data used by the client (for example, an SQL query).
TiClientName	F_String	The registered name of the client that created the inset.
TiClientSource	F_String	The name that appears as the source in the Text Inset Properties dialog box.
TiClientType	F_String	The name that appears as the source type in the Text Inset Properties dialog box.
TiIsUnresolved	F_Int	True if the inset is unresolved. A client should set this property to True if it is unable to resolve the inset.

FMTextInset_Flow

Details

FMTextInset_Flow objects represent text insets for text imported from a FrameMaker document.

See also

"FMObject" on page 166, "FMTextInset" on page 217.

Properties

Property:	Type:	Description:
TiFlowName	F_String	The name of the imported flow if TiMainFlow is False. If the source file is an edition, TiFlowName is set to Macintosh edition.
TiFlowPageSpace	F_Int	The type of pages the imported flow is on: mFV_BODY_PAGE mFV_REFERENCE_PAGE

Property:	Type:	Description:
TiFormat	F_Int	Source of the imported text's format: mFV_SourceDoc: the text is formatted with formats from the source document mFV_PlainText: the text is formatted as plain text mFV_EnclosingDoc: the text is formatted with formats from the document into which it is imported
TiMainFlow	F_Int	True if the inset text is imported from the main flow of the source document.
TiRemoveOverrides	F_Int	True if page breaks are removed from the text when TiFormat is set to mFV_EnclosingDoc.
TiRemovePageBreaks	F_Int	True if format overrides are removed from the text when TiFormat is set to mFV_EnclosingDoc.

FMTextInset_Text

Details

FMTextInset_Text objects represent text insets for text imported from a text file.

See also

"FMObject" on page 166, "FMTextInset" on page 217.

Properties

Property:	Type:	Description:
TiEOLisEOP	F_Int	True if line ends in the imported text file are treated as paragraph ends.
TiTextEncoding	F_String	The ImportHintString for the text inset. If this is not a FMTextInset_Text or FMTextInset_TextTable, the string is null.

FMTextInset_TextTable

Details

FMTextInset_TextTable represent text insets for text imported from a delimited text file. The delimiters indicate records and fields.

See also

"FMObject" on page 166, "FMTextInset" on page 217.

Properties

Property:	Type:	Description:
TiByRows	F_Int	True if each paragraph in the imported text is converted to a row of table cells; False if each paragraph in the imported text is converted to a table cell.
TiHeadersEmpty	F_Int	True if the imported text is not used to fill the heading rows.
TiNumCols	F_Int	If TiByRows is False, the number of columns in the table.
TiNumHeaderRows	F_Int	The number of heading rows in the table.
TiNumSeparators	F_Int	If TiSeparator specifies a space, the number of spaces used as a separator to parse the text into table cells.
TiSeparator	F_String	If TiByRows is True, a string specifying a separator, such as a tab, used to parse the text into table cells.
TiTblTag	F_String	The table format tag of the imported table.
TiTextEncoding	F_String	The ImportHintString for the text inset. If this is not a FMTextInset_Text or FMTextInset_TextTable, the string is null.

FMTextLine

Details

FMTextLine objects represent instances of the text you can add to a graphic via the FrameMaker Text drawing tool. To get the text from a text line, use the GetText method that the text line inherits from FMObject.

See also

"FMObject" on page 166, "FMGraphic" on page 155.

Properties

Property:	Type:	Description:
BasePointX	F_Metric	Horizontal placement of text line base point relative to left side of frame.
BasePointY	F_Metric	Vertical placement of text line base point relative to top of frame.

Property:	Type:	Description:
Language	F_Int	Obsolete, but still functional. Hyphenation and spell-checking language to use for the first character in the text line. Because language can be specified for a character format, you should get FP_Language from the FMCharacterFormat objects in the text of the text line object. The possible values for a text line are: mFV_LANG_BRAZILIAN mFV_LANG_BRITISH mFV_LANG_CANADIAN_FRENCH mFV_LANG_CATALAN mFV_LANG_DANISH mFV_LANG_DUTCH mFV_LANG_ENGLISH mFV_LANG_FINNISH mFV_LANG_FRENCH mFV_LANG_GERMAN mFV_LANG_ITALIAN mFV_LANG_NOLANGUAGE mFV_LANG_NORWEGIAN mFV_LANG_NYNORSK mFV_LANG_PORTUGUESE mFV_LANG_SPANISH mFV_LANG_SWEDISH mFV_LANG_SWISS_GERMAN mFV_LANG_JAPANESE mFV_LANG_TRADITIONAL_CHINESE mFV_LANG_SIMPLIFIED_CHINESE mFV_LANG_KOREAN

Property:	Type:	Description:
TextLineType	F_Int	Type of text line: mFV_TEXTLINE_LEFT mFV_TEXTLINE_RIGHT mFV_TEXTLINE_CENTER mFV_TEXTLINE_MATH

FMUnanchoredFrame

Details

FMUnanchoredFrame objects represent the unanchored frames you can draw in a document using the FrameMaker drawing tools.

See also

"FMObject" on page 166, "FMGraphic" on page 155.

Properties

Property:	Type:	Description:
FirstGraphicInFrame	FMGraphic	First object in the frame (backmost object).
LastGraphicInFrame	FMGraphic	Last object in the frame (frontmost object).
Name	F_String	If a reference frame, the frame's name.
PageFramePage	FMPage	If the unanchored frame is a page frame, the page that it belongs to (FMBodyPage, FMHiddenPage, FMMasterPage, or FMReferencePage).

FMVariable

Details

FMVariable objects represent instances of variables in a document.

See also

"FMObject" on page 166.

Properties

Property:	Type:	Description:
Element	FMElement	If the variable is in a FrameMaker+SGML document, the ID of the element associated with the variable.
Locked	F_Int	True if the variable is included in a text inset that gets its formatting from the source document. The variable is not affected by global formatting performed on the document.
NextVarInDoc	FMVariable	Next variable instance in the document.
TextRange	udtTextRange	The text range the variable instance encompasses.
Unique	F_Int	The variable's UID.
VarFmt	FMVariableFormat	The variable instance's format.

FMVariableFormat

Details

FMVariableFormat objects represent the variable formats that are stored with a document. You can create and delete user variables, but you cannot delete system variables from the document.

See also

"FMObject" on page 166, "FMGraphic" on page 155.

Properties

Property:	Type:	Description:
Fmt	F_String	The variable format definition; the building blocks and text strings used to create a variable instance with the variable format.
Name	F_String	The variable format's name.
NextVarFmtInDoc	FMVariableFormat	Next variable format in the document's list of variable formats.

Property:	Type:	Description:
SystemVar	F_Int	The variable format's type. mFV_VAR_USER_VARIABLE: a user-defined variable format. The following types specify system variable formats: mFV_VAR_CURRENT_PAGE_NUM mFV_VAR_PAGE_COUNT mFV_VAR_CURRENT_DATE_LONG mFV_VAR_CURRENT_DATE_SHORT mFV_VAR_MODIFICATION_DATE_LONG mFV_VAR_MODIFICATION_DATE_SHORT mFV_VAR_CREATION_DATE_LONG mFV_VAR_CREATION_DATE_SHORT mFV_VAR_FILE_NAME_LONG mFV_VAR_FILE_NAME_SHORT mFV_VAR_HEADER_FOOTER_1 through mFV_VAR_HEADER_FOOTER_12 mFV_VAR_TABLE_CONTINUATION mFV_VAR_TABLE_SHEET

3

4 Event Reference

This chapter lists the FrameAC events that your applications may handle. These events provide you the means to respond to:

- User commands — the OnApiCommand event responds to menu and command choices
- Hypertext commands — the OnApiMessage event responds to FrameMaker hypertext links that have a special syntax
- User actions — the OnApiNotify event responds to a wide range of actions that can be performed in the FrameMaker session

Each event description includes an example of how to use the event in your code.

OnApiCommand

Synopsis

```
OnApiCommand(command As Long)
```

Arguments

command	A value that identifies which menu command was chosen..

Details

This event is fired when the user selects a menu command that was defined by the plug-in or controller application. Each menu command includes a Long value that uniquely identifies the command within the process space of the plug-in. This event receives that value — depending on the command, the plug-in can branch into the appropriate routines.

Examples

The following code shows how to respond to menu commands. It includes the following sections and Sub routines:

- General declarations — Define command constants and global variables
  Note the use of command constants. This is a preferred way to refer to the commands in your code. Also note that the values are 901 and 902. All FrameAC plug-ins share events. For this reason, you must provide command numbers that are unique for all FrameAC plug-ins that are running. You must establish your own conventions to ensure these values are unique.
- Initialise — Establishes the connection with FrameMaker, and displays the menus
  This is fairly straight-forward. It assigns the current session to a global variable, and then calls the AddMenus Sub routine.

- AddMenus — Does the work of defining and posting the menus
  Note the use of the command constants
- OnApiCommand — Responds to the menu commands
  This event receives the current command as an argument, and uses a Select Case statement to branch into the appropriate actions.

```
Option Explicit
'********************
' Menu Cmd Constants
'********************
Const HELLO_WORLD_CMD As Long = 901
Const GOODBYE_CRUEL_WORLD_CMD As Long = 902
Public WithEvents theSession As FMSession
```

```
Public Sub Initialise(iSession As Object, cmdFirst As Long)
Dim FACAuthorise As FrameAuthorise
     Set FACAuthorise = New FrameAuthorise
     If iSession.Initialise(FACAuthorise) = 1 Then
          Set theSession = iSession
          AddMenus
     End If
End Sub
```

```
Private Sub AddMenus()
Dim oMainMenu As FMMenu, oMMenu As FMMenu
Dim oNewCommand As FMCommand
     ' Get the framemaker main menu
     Set oMainMenu = theSession.GetNamedObject("!MakerMainMenu", mFO_Menu, Nothing)

     ' Check if the new menu already exists
     Set oMMenu = theSession.GetNamedObject("EventResponder", mFO_Menu, Nothing)
     If oMMenu Is Nothing Then
          ' menu doesn't exist so now add it
          Set oMMenu = theSession.DefineMenu("EventResponderMENU", "EventResponder")
          oMainMenu.AddMenuToMenu oMMenu

          ' Now that we have the menu we can add commands to it
```

```
        ' create the command object and add to menu
        Set oNewCommand = theSession.DefineCommand _
             (HELLO_WORLD_CMD, "EventResponderHelloWorld", "Hello World", "")
        oNewCommand.AddCommandToMenu oMMenu

        Set oNewCommand = theSession.DefineCommand _
           (GOODBYE_CRUEL_WORLD_CMD, "EventResponderGoodbyeWorldC", "Goodbye", "")
        oNewCommand.AddCommandToMenu oMMenu
    End If
End Sub
```

```
Private Sub theSession_OnApiCommand(ByVal cmd As Long)
Dim oDoc As FMDocument

    ' Branch to the appropriate command
    Select Case cmd
        Case HELLO_WORLD_CMD
            theSession.Alert "Hello World", mFF_ALERT_CONTINUE_NOTE
        Case GOODBYE_CRUEL_WORLD_CMD
            theSession.Alert "Goodbye Cruel World", mFF_ALERT_CONTINUE_NOTE
    End Select
End Sub
```

OnApiMessage

Synopsis

```
OnApiMessage(message As String,
docId As Long,
objId As Long)
```

Arguments

message	The content of the message string
docId	The ID of the document containing the hypertext marker
objId	The ID of the hypertext marker

Details

This event is fired when the user clicks a hypertext link in the FrameMaker document, and that link has the following syntax:

```
message FrameAC [optional message string]
```

Note that a message event is sent to all FrameAC plug-ins. For that reason, the message string should indicate to which plug-in the event is directed.

Examples

The following code shows how the OnApiMessage event responds to a message hypertext command. It displays the message string and the name of the document that contains the hypertext marker.

For a practical response to a message event, you should establish messages that identify which plug-in will respond, and exactly what the response should be. From there, you can test the message string, and branch into the appropriate routines.

This code assumes the current FrameMaker session was assigned to the global variable, `theSession`.

```
Private Sub theSession_OnApiMessage(ByVal message As String, _
                                    ByVal docId As Long, _
                                    ByVal objId As Long)
Dim oDoc As FMDocument
     theSession.Alert message, mFF_ALERT_CONTINUE_NOTE
     Set oDoc = theSession.GetObject(docId, Nothing)
     theSession.Alert oDoc.Name, mFF_ALERT_CONTINUE_NOTE
End Sub
```

OnApiNotify

Synopsis

```
OnApiNotify(notification As Long,
docId As Long,
sparam As String,
iparam As Long)
```

Arguments

notification — A constant that indicates the action and the notification point. Se the documentation for the SetNotification method for a list of the constants.

docId — The ID of the active document when the event occurs. For filters, the document into which the filter should import its data — if this is zero, the filter must create a new document.

sparam	The string, if any, associated with the notification. For example, notifications for open operations specify the pathname of the file to open. If the notification is for text entry, this specifies the text the user typed.
iparam	An integer passed with the notification. For example, if notification is for `mFA_Note_PreFunction` or `mFA_Note_PostFunction`, this is the f-code for the command.

Details

Important: For book-wide commands, the FrameMaker product posts an `mFA_NotePreFunction` and `mFA_NotePostFunction` notification for the book file, and for each document in the book. When trapping book-wide functions, you should check the value of docId to determine whether it indicates a document or the active book.

For example, if you search a book with two documents in it, the FrameMaker product posts the following function notifications:
`mFA_Note_PreFunction` (start searching book)
`mFA_Note_PreFunction` (start searching first document)
`mFA_Note_PostFunction` (stop searching first document)
`mFA_Note_PreFunction` (start searching second document)
`mFA_Note_PostFunction` (stop searching second document))
`mFA_Note_PostFunction` (stop searching book)

This event is triggered for any notifications that have been requested. Note FrameAC shares the notification requests that are made by all the FrameAC plug-ins. However, you set up the OnApiNotify event to respond only to the notifications that are of interest to you.

For each notification event, FrameAC passes `sparam` and `iparam` — a string and an integer that you can use to further qualify a notification.

The following table shows the values of `sparam` for specific notifications:

Notifications	sparam value
All Open, Save, Print, and Close notifications for documents and books	The complete pathname of the document or book.
`mFA_Note_FilterIn`	The pathname of the file to filter .
`mFA_Note_FilterOut`	The pathname of the file the filter will create.
`mFA_Note_FilterFileToTile`	A pointer to an `F_FilterArgsT` structure.
`mFA_Note_PreDistill` `mFA_Note_PostDistill`	When this notification occurs as a result of saving a document or book as PDF, `sparm` contains the complete pathname of thePostScript file that was generated from the document or book.

Notifications	sparam value
mFA_Note_PreFunction mFA_Note_PostFunction	If the user typed text, the text. If the user applied a paragraph or character format or a font family, the name of the format or font family.
mFA_Note_PreChangeElement mFA_Note_PostChangeElement	The element tag of the changed structural element.
mFA_Note_PreGenerate mFA_Note_PostGenerate	The book's filename. If the book is untitled, NULL.
mFA_Note_PreHypertext	The string of the hypertext command.
mFA_Note_PostHypertext	Null
mFA_Note_PreImport mFA_Note_PostImport	The name of the imported file.
mFA_Note_PreInsertElement	The element tag of the inserted structural element.
mFA_Note_PostInsertElement	The element tag of the inserted structural element.
mFA_Note_PreOpenBook mFA_Note_PostOpenBook	The complete pathname of the book file. If the book is untitled, NULL.
mFA_Note_PreSetAttrValue mFA_Note_PostSetAttrValue	The attribute name.
mFA_Note_PreWrapElement	The element tag of the wrapped structural element.
mFA_Note_PostWrapElement	The element tag of the wrapped structural element.
All other notifications	NULL.

The following table shows the values of iparam for specific notifications:

Notifications	iparam value
mFA_Note_FilterIn	Id of the active document, if there is one.
mFA_Note_FilterOut	Id of the document to filter, or the dcument containing the graphic to filter.
mFA_Note_FilterFileToTile	For import filters, the Id of the active document, if there is one. For export filters, the Id of the document to filter, or the dcument containing the graphic to filter.
mFA_Note_PreFunction mFA_Note_PostFunction	The f-code for the command the user invoked. For a list of f-code constants, see the fcodes.h header file provided with the FDK.

Notifications	iparam value
mFA_Note_UpdateAllClientTi	One of the following flags to indicate which text insets are to be updated: FV_UpdateAllAutomaticClientTi indicates all insets with the FP_TiAutomaticUpdate property set to True. FV_UpdateAllManualClientTi indicates all insets with the FP_TiAutomaticUpdate property set to False. FV_UpdateAllClientTi indicates all insets regardless of the setting for the FP_TiAutomaticUpdate property.
mFA_Note_UpdateClientTi	The ID of the text inset.
mFA_Note_PreHypertext mFA_Note_PostHypertext	The ID of the hypertext object that was activated. If the hypertext message is for an FDK client, `iparm` is the same as the ID passed to the `objId` parameter of the client's `F_ApiMessage()` callback.
mFA_Note_PreGoToXrefSrc mFA_Note_PostGoToXrefSrc	The ID of the associated cross-reference
mFA_Note_PreMouseCommand mFA_Note_PostMouseCommand	The 8 high bits specify the number of mouse clicks minus one. The next 8 bits indicate the modifier key used:`FF_SHIFT_KEY`, `FF_CONTROL_KEY`, `FF_ALT_KEY`, or `FF_CMD_KEY` The 16 low bits specify the mouse action: for example, `FF_TEXT_SEL` if the user is selecting text; `FF_TEXT_EXT` if the user is extending an existing selection; or `OBJ_SEL` if the user is selecting an object. For a complete list of the constants for mouse actions, see `/* Mouse actions */` in the `fapidefs.h` header file provided with the FDK.
mFA_Note_PreInsertElement mFA_Note_PostInsertElement	The ID of the inserted structural element.
mFA_Note_PreWrapElement mFA_Note_PostWrapElement	The ID of the newly created structural element.
mFA_Note_PreDragElement	The ID of the structural element that is cut if the operation is completed.
mFA_Note_PostDragElement	The ID of the structural element that is pasted if the operation is completed.
mFA_Note_PreSetAttrValue mFA_Note_PostSetAttrValue	The ID of the element for which the attribute is set.
mFA_Note_PreImportElemDefs	The ID of the document from which element definitions are imported.

Notifications	iparam value
mFA_Note_PreOpenBook mFA_Note_PostOpenBook	The ID of the book being opened.
mFA_Note_PreOpenDoc mFA_Note_PostOpenDoc mFA_Note_PreOpenSGML mFA_Note_PostOpenSGML mFA_Note_PreOpenXML mFA_Note_PostOpenXML	The ID of the document being opened.
mFA_Note_PreSaveBook mFA_Note_PostSaveBook	The ID of the book being saved.
mFA_Note_PreSaveDoc mFA_Note_PostSaveDoc mFA_Note_PreSaveSGML mFA_Note_PostSaveSGML mFA_Note_PreSaveXML mFA_Note_PostSaveXML	The ID of the document being saved.
mFA_Note_PreSaveAsPDFDialog mFA_Note_PostSaveAsPDFDialog	When this notification occurs as a result of the user choosing to save as PDF in the Save As dialog box, the value is non-zero. When this notification is the result of the API saving as PDF, the value is zero.
All other notifications	0.

Examples

The following code includes two Sub routines:

- Initialise — Establishes a connection with the FrameMaker session, then requests three notification points
- OnApiNotify — Handles the notification event, and branches into the appropriate routines, depending on the notification code.

```
Public Sub Initialise(iSession As Object, cmdFirst As Long)
' Function called by frame on frame initialization

' authorise FrameAC
Dim FACAuthorise As FrameAuthorise
      Set FACAuthorise = New FrameAuthorise
      If iSession.Initialise(FACAuthorise) = 1 Then
            ' communication initialised ok
            Set theSession = iSession
```

```
        ' Set notification points
        iSession.SetNotification mFA_Note_PrePrint, True
        iSession.SetNotification mFA_Note_ConsoleMessage, True
        iSession.SetNotification mFA_Note_PreHypertext, True
    End If
End Sub
```

```
Private Sub theSession_OnApiNotify(ByVal notification As Long, _
                                   ByVal docId As Long, _
                                   ByVal sparam As String, _
                                   ByVal iparam As Long)

    Select Case notification
        Case mFA_Note_PrePrint
            theSession.Alert "Printing!", mFF_ALERT_CONTINUE_NOTE
            'theSession.FA_errno = mFR_CancelOperation
        Case mFA_Note_ConsoleMessage
            theSession.Alert sparam, mFF_ALERT_CONTINUE_NOTE
        Case mFA_Note_PreHypertext
            theSession.Alert _
                "mFA_Note_PostHypertext - " + sparam + ": " + Str(iparam), _
                mFF_ALERT_CONTINUE_NOTE
    End Select
End Sub
```

4

5 Method Reference

This chapter lists the FrameAC methods alphabetically. If you know a method that you want to use, but need more information about how to use it, you can look it up here. Each method description includes an example of how to call the method in code. Some methods are used in expanded examples that appear in Chapter 4, "Event Reference," and in Chapter 2, "FrameAC Tutorial."

AddCols

Add columns to a table.

Synopsis

```
AddCols(
RefColumn As Long,
Direction As Long,
Count As Long) As Long
```

Arguments

RefColumn. The column at which to start adding columns. Table columns are numbered from left to right, starting with column `0`.

Direction. Whether to add to the right or the left. Can be one of `mFV_Left` or `mFV_Right`

Count. The number of columns to add

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`FE_BadParameter`	The function call specified an invalid parameter
`FE_BadOperation`	The function call specified an illegal operation

Examples

The following code gets the currently selected table and adds one column to the left of the first column.

```
Dim oDoc As FMDocument
Dim oTbl As FMTable
```

```
Set oDoc = mySession.ActiveDoc
Set oTbl = oDoc.SelectedTbl

oTbl.AddCols 0, mFV_Left, 1
```

AddCommandToMenu

Adds a command to a menu.

Synopsis

```
AddCommandToMenu(MenuToAddTo As FMMenu) As Long
```

Arguments

MenuToAddTo. The menu object that will display the command.

Details

To add a command to a menu, you must get or create the command object. Then you must get or create the menu object you will add the command to. To get a menu object, you can use the session's GetNamedObject method, passing the menu's Name parameter for the object's name. To see the menu names for your installation of FrameMaker, see the file, `$PRODUCT_DIR\fminit\maker\menus.cfg`.

You should be aware that users can create custom menu files that override the default menus. For more information, see your FrameMaker installation documentation.

> **Important:** Commands you add to menus persist through the FrameMaker session. For example, if you add a command via a controller application, that command will persist even after the user quits the controller application. For that reason, you should be sure to remove any menu commands you added if your processing context no longer supports them.

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_WrongProduct`	Current FrameMaker product doesn't support this operation
`mFE_BadOperation`	Parameters specify an invalid operation
`mFE_NotMenu`	`MenuToAddTo` doesn't specify a menu
`mFE_BadParameter`	Parameter has an invalid value
`mFE_SystemError`	System error

Examples

The following example adds a command to the File menu.

```
Dim oMenu As FMMenu, oNewCommand As FMCommand

'Get the  File menu.
Set oMenu = mySession.GetNamedObject("FileMenu", mFO_Menu, Nothing)

'Now create the command object and add it to the File menu.
Set oNewCommand = mySession.DefineCommand(9902, "MyCommand", "My Command", "")
oNewCommand.AddCommandToMenu oMenu
```

AddMenuToMenu

Adds a FrameMaker product menu or a menu that you have created to another menu or menu bar.

Synopsis

```
AddMenuToMenu(MenuToAdd As FMMenu) As Long
```

Arguments

MenuToAdd. The menu you want to add to the current menu object.

Details

To add a menu to a menu, you must get or create the new menu object and add commands to it. Then you must get or create the parent menu object and call the method from this parent.

To get a menu object, you can use the session's GetNamedObject method, passing the menu's Name parameter for the object's name. To see the menu names for your installation of FrameMaker, see the file, `$PRODUCT_DIR\fminit\maker\menus.cfg`.

You should be aware that users can create custom menu files that override the default menus. For more information, see your FrameMaker installation documentation.

> **Important:** Commands and menus you add will persist through the FrameMaker session. For example, if you add a menu via a controller application, that menu will persist even after the user quits the controller application. For that reason, you should be sure to remove any menus or commands you added if your processing context no longer supports them.
>
> Your menu appears only on the menu bar you specify. For example, if you only add a menu to the `!MakerMainMenu` menu bar, the menu will not appear if the user switches to quick menus. For your menu to appear after the user has switched to quick menus, you must also add it to `!QuickMakerMainMenu`.

The following table lists the types of menus you can add a menu to and how the FrameMaker product implements the added menu:

Type of menu or menu bar you are adding a menu to	How the FrameMaker product implements the added menu	Where the FrameMaker product adds the menu
Menu bar	Pull-down menu	At the right of the menu bar
Pull-down menu	Pull-right menu	At the bottom of the pull-down menu
Pop-up menu	Pull-right menu	At the bottom of the pop-up menu
Pull-right menu	Pull-right menu	At the bottom of the pull-right menu

To change a menu's position on a menu or menu bar after you add it, set its PrevMenuItemInMenu and NextMenuItemInMenu properties.

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_WrongProduct`	Current FrameMaker product doesn't support this operation
`mFE_BadOperation`	Parameters specify an invalid operation
`mFE_NotMenu`	`MenuToAddTo` doesn't specify a menu
`mFE_BadParameter`	Parameter has an invalid value
`mFE_SystemError`	System error

Examples

The following code creates a menu and adds a command to it. The code then adds the new menu to the File men.

```
Dim oMenu As FMMenu, oNewMenu As FMMenu
Dim oNewCommand As FMCommand

Set oNewMenu = mySession.DefineMenu("MyNewMenu", "My New Menu")
Set oNewCommand = mySession.DefineCommand(9902, "MyCommand", "MyCommand", "")
oNewCommand.AddCommandToMenu oNewMenu

Set oMenu = mySession.GetNamedObject("FileMenu", mFO_Menu, Nothing)
oMenu.AddMenuToMenu oNewMenu
```

AddRows

Method Short description.

Synopsis

```
AddRows(RefRow As FMRow,
Direction As Long,
Count As Long) As Long
```

Arguments

RefRow.	The row from which to start adding more rows.
Direction.	The direction from the reference row to start adding rows.
Count.	The number of rows to add

Details

The following table lists the constants you can specify for `Direction`:

Direction constant	Meaning
`mFV_Above`	Add rows above the row specified by `RefRow`. The added rows are the same type as the row specified by `RefRow`.
`mFV_Below`	Add rows below the row specified by `RefRow`. The added rows are the same type as the row specified by `RefRow`.
`mFV_Body`	Add body rows at the bottom of the existing body rows (`RefRow` is ignored).
`mFV_Footing`	Add footing rows at the bottom of the existing footing rows (`RefRow` is ignored).
`mFV_Heading`	Add heading rows at the bottom of the existing heading rows (`RefRow` is ignored).

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_WrongProduct`	Current FrameMaker product doesn't support this operation
`mFE_BadObjId`	Invalid row object
`mFE_BadOperation`	Parameters specify an invalid operation
`mFE_BadParameter`	Parameter has an invalid value

Examples

The following code adds two rows above the first body row in the selected table.

```
Dim oDoc As FMDocument
Dim oTbl As FMTable, oRow As FMRow

Set oDoc = mySession.ActiveDoc
Set oRow = oDoc.SelectedTbl.FirstRowInTbl

'Make sure oRow is the first body row in the table.
Do Until oRow Is Nothing
     If oRow.RowType = mFV_ROW_BODY Then
          Exit Do
     End If
     Set oRow = oRow.NextRowInTbl
Loop

'Add two rows above oRow.
oTbl.AddRows oRow, mFV_Above, 2
```

AddText

Inserts text into a paragraph or text line.

Synopsis

```
AddText(Loc,
Text As String)
```

Arguments

Loc	A text location in a document at which to add the text
Text	The text to add

Details

The text you specify for `text` must use the FrameMaker product character set. To add special characters, you must specify octal (\) or hexadecimal (\x) sequences. The following table lists some of these sequences.

Special character	Hexadecimal representation	Octal representation
>	\x3e	\76
" (straight double quotation mark)	\x22	\42
“ (left double quotation mark)	\xd2	\322

Special character	Hexadecimal representation	Octal representation
” (right double quotation mark)	\xd3	\323

For a complete list of the characters in the FrameMaker product character set and the corresponding hexadecimal codes, see your Frame product user's manual.

Returns

Nothing. On success, sets `mFE_Success` to `FA_errno`. On failure it sets `FA_errno` to one of the following error codes:

Error code	Meaning
`mFE_OffsetNotFound`	The offset specified for the text location couldn't be found in the specified text object.
`mFE_ReadOnly`	The specified document is read-only.
`mFE_BadSelectionForOperation`	The location that textLocp specifies is invalid. For example, it is inside a variable or outside the highest level element in a structured FrameMaker document.

Examples

The following code gets the current text selection in a document and adds a string of text at the beginning of it.

```
Dim oDoc As FMDocument
Dim err As Long

Set oDoc = mySession.ActiveDoc
oDoc.AddText oDoc.TextSelection.beg, "This is my new text."
```

Alert

Displays an alert box with a message. Depending on the constant you specify for `type`, it displays either OK and Cancel buttons, Yes and No buttons, or a Continue button.

Synopsis

```
Alert(message As String,
type As Long) As Long
```

Arguments

message	The message that appears in the dialog box. Strings longer than 255 characters are truncated.
type	The dialog box type. See below for a list of types.

Details

The alert box type is specified by the following constants:

type constant	Type of dialog box displayed
mFF_ALERT_OK_DEFAULT	Displays OK and Cancel buttons; OK is the default
mFF_ALERT_CANCEL_DEFAULT	Displays OK and Cancel buttons; Cancel is the default
mFF_ALERT_CONTINUE_NOTE	Displays OK button
mFF_ALERT_CONTINUE_WARN	Displays OK button with a warning indication
mFF_ALERT_YES_DEFAULT	Displays Yes and No buttons; Yes is the default
mFF_ALERT_NO_DEFAULT	Displays Yes and No buttons; No is the default

Returns

`0` if the user clicked OK, Continue, or Yes; `-1` if the user clicked Cancel or No.

Examples

The following code displays an alert box with OK and Cancel buttons. It then tests the user's response and branches approprately.

```
Dim resp As Long
resp = mySession.Alert("Do you want to continue?", mFF_ALERT_OK_DEFAULT)
If resp = 0 Then
     'Perform the continued action.
     mySession.Alert "CONTINUE!", mFF_ALERT_CONTINUE_NOTE
Else
     'Do not continue.
     mySession.Alert "QUIT!", mFF_ALERT_CONTINUE_NOTE
End If
```

ApplyPageLayout

Applies the layout of one page to another page.

Synopsis

```
ApplyPageLayout(SourcePage As FMPage) As Long
```

Arguments

Source Page The page with the layout to apply

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_WrongProduct	Current FrameMaker product doesn't support this operation
mFE_BadOperation	Parameters specify an invalid operation
mFE_BadParameter	Parameter has an invalid value
mFE_SystemError	System error

Examples

The following code applies the format from the master page named "First" to the current page in the document.

```
Dim oDoc As FMDocument
Dim oPg As FMBodyPage, oMPg As FMMasterPage

Set oDoc = mySession.ActiveDoc
Set oPg = oDoc.CurrentPage

'Get the master page named "First".
Set oMPg = oDoc.FirstMasterPageInDoc
Do Until oMPg Is Nothing
     If oMPg.Name = "First" Then
          Exit Do
     End If
     Set oMPg = oMPg.PageNext
Loop
If oMPg Is Nothing Then
     mySession.Alert "No MPg named First", mFF_ALERT_CONTINUE_NOTE
     Exit Sub
End If

'Apply master page's layout to the current page.
oPg.ApplyPageLayout oMPg
```

BailOut

Prepares your ActiveX plug-ins to free up resources when they return control back to FrameMaker.

Synopsis

```
BailOut() As Long
```

Details

A plug-in does not bail out until after the current event returns control to FrameMaker. For this reason, you can place the call to BailOut anywhere in a specific process—it's usually best to make the call at the beginning of the process.

After your client has bailed out, the FrameMaker product still processes events that affect it. Menus your client created are still valid. If your client has requested notification for particular events, the FrameMaker product continues to monitor those events. It also monitors apiclient hypertext commands that specify your client's name. If these events occur, the FrameMaker product restarts your client by calling its Initialise event with initialization set to mFA_Init_Subsequent.

Important: If your plug-in bails out, it loses all its global variable values.

Examples

The following code calls BailOut immediately after successfully initialising a session.

```
Public Sub Initialise(iSession As Object, cmdFirst As Long)


Dim FACAuthorise As FrameAuthorise
Set FACAuthorise = New FrameAuthorise
If iSession.Initialise(FACAuthorise) = 1 Then
     iSession.BailOut
...
```

CallClient

Calls an existing plug-in from your code. For example, you can call the FrameMaker installed plug-ins such as the structure generator or the table sorting plug-in from your code.

Synopsis

```
CallClient(ClientName As String,
Argument As String) As Long
```

Arguments

ClientName	The registered name of the target client. For more information on how FDK plug-ins are named and which plug-ins are included with FrameMaker, see the FrameMaker FDK documentation.
Argument	string that is passed to the target client.

Details

CallClient actually calls FrameAC — FrameAC then passes the call to any FrameAC plug-ins that requested the OnApiNotify event. To receive the call, FrameAC must be correctly registered in the maker.ini file, and the individual plug-in must have requested the `mFA_Note_ClientCall` notification (see "SetNotification" on page 363 for more information).

When calling an FDK plug-in, the target plug-in must be properly implemented to receive the notification. For more information, refer to the developer of the FDK plug-in, or see the FrameMaker FDK documentation.

A number of features in the FrameMaker product are implemented via FDK clients. To execute these functions in a client of your own, you call them via CallClient. For descriptions of how to call these clients from your own code, see the FrameMaker FDK documentation.

Returns

`mFE_Success` if it succeeds, or the value specified by the taget plug-in's last call to `F_ApiReturnValue()` (for FDK plug-ins, only).

If CallClient fails it assigns one of the following error codes to the session's FA_Errno property:

Error code	Meaning
`mFE_NameNotFound`	There is no client with the specified name in the current FrameMaker product session
`mFE_BadParameter`	For the FrameMaker TableSort client only: One of the arguments is invalid. For example, you gave a value for the sort key that is greater than the number of columns or rows in the current table selection, or you have no table cells selected.

Examples

The following code sorts the selected rows of a table in ascending order. Note that all the functionality comes from the table sorting plug-in—the table sorting plug-in identifies the table and the rows to sort based on the user's selection. This code shows that all you need to call an existing plug-in is a valid connection to a FrameMaker session.

```
Option Explicit

Public myAuth As New FrameAuthorise

Public myEx As New FrameEx

Public mySession As FMSession


Private Sub MyButton_Click()


Set mySession = myEx.Session
```

```
If mySession.Initialise(myAuth) = 1 Then
mySession.CallClient "TableSort", "row nocase 0 ascending -1 descending
     -1 descending"
End If
End Sub
```

CenterOnText

Centres a range of text so the text appears within the viewing area of the document window.

Synopsis

```
CenterOnText(textRange) As Long
```

Arguments

textRange The range of text to centre

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_BadRange	The specified text range is invalid
mFE_NotTextObject	One of the objects specified for the text range is not a paragraph or text line
mFE_OffsetNotFound	The offset specified for the text location couldn't be found in the specified paragraph or text line

Examples

The following code centres the text selection or insertion point.

```
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
oDoc.CenterOnText oDoc.TextSelection
```

Clear

Deletes the current selection from a document.

Synopsis

```
Clear(flags As Long) As Long
```

Arguments

Flags	Bit field that specifies how to handle interactive alerts. For the default action, specify 0.

Details

If you specify 0 for flags, F_ApiClear() suppresses any interactive alerts or warnings that arise, leaves the selected table cells empty, and deletes hidden text. You can also OR the following values into flags:

flags constant	Meaning
mFF_INTERACTIVE	Prompt user with dialog or alert boxes that arise
mFF_CUT_TBL_CELLS	Remove cleared table cells
mFF_VISIBLE_ONLY	Clear only the visible portion of the selection
mFF_DONT_DELETE_HIDDEN_TEXT	Don't delete hidden text

The mFF_INTERACTIVE flag takes precedence over other flags. So, if you specify mFF_INTERACTIVE | mFF_DONT_DELETE_HIDDEN_TEXT and the selection contains hidden text, FrameMaker allows the user to choose whether to delete the hidden text.

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
FE_BadParameter	The function call specified an invalid parameter
FE_BadOperation	The function call specified an illegal operation

Examples

The following code changes the text selection to be the complete paragraph that contains the start of the selection. It then deletes the paragraph.

```
Dim oDoc As FMDocument
Dim oPgf As FMParagraph
Dim f_tRange As udtTextRange
Dim err As Long

Set oDoc = mySession.ActiveDoc
f_tRange = oDoc.TextSelection
Set oPgf = oDoc.MainFlowInDoc.FirstTextFrameInFlow.FirstPgf
err = 0

'Get the paragraph object that matches the first
```

```
'objId in the text range.
Do Until err = 1
     If f_tRange.beg.objId = oPgf.Id Then
          Exit Do
     End If
     If oPgf Is Nothing Then
          err = 1
     End If
     Set oPgf = oPgf.NextPgfInFlow
Loop
If err = 1 Then
     Exit Sub
End If

'Now extend the selection in both directions.
Set oPgf = oPgf.NextPgfInFlow
If oPgf Is Nothing Then
     Exit Sub
End If
f_tRange.end.objId = oPgf.Id
f_tRange.beg.offset = 0
f_tRange.end.offset = 0
oDoc.TextSelection = f_tRange

'Finally, delete the paragraph.
oDoc.Clear mFF_INTERACTIVE
```

ClearAllChangebars

Clears change bars from a specified document. It executes the same command as clicking the Clear All Change Bars box in the Change Bars dialog box.

Synopsis

```
ClearAllChangebars() As Long
```

Details

Method description goes here.

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_WrongProduct	Current Frame product doesn't support this operation
mFE_SystemError	System error

Examples

The following code removes the change bars from the current document.

```
Set oDoc = mySession.ActiveDoc
oDoc.ClearAllChangebars
```

Close

Closes a document or book.

Synopsis

```
Close(flags As Long) As Long
```

Arguments

Flags — Specifies whether to abort or to close open documents or books if they have unsaved changes. Can be one of mFF_CLOSE_MODIFIED or 0. Set the mFF_CLOSE_MODIFIED flag to close open documents and books regardless of their state.

Details

If there are unsaved changes in the document or book and you set mFF_CLOSE_MODIFIED for the flags argument, Close abandons the changes and closes the file anyway. If you set flags to 0, Close aborts the Close operation and returns mFE_DocModified.

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_DocModified	The document was modified and flags was set to 0

Examples

The following code closes the active document, with a warning if the document has unsaved changes.

```
Dim oDoc As FMDocument
Dim resp As Long
```

```
Set oDoc = mySession.ActiveDoc
resp = 0
If oDoc.DocIsModified Then
     resp = mySession.Alert("Close without saving changes?", mFF_ALERT_OK_DEFAULT)
End If

If Not resp Then
     oDoc.Close mFF_CLOSE_MODIFIED
End If
```

Compare

Compares two documents or two books.

Synopsis

```
Compare(OriginalDoc As FMDocument,
flags As Long,
insertCondTag As String,
deleteCondTag As String,
replaceText As String,
CompareThreashold As Long)
```

Arguments

OriginalDoc	The older document or book to compare against.
Flags	Bit flags that specify how to generate the summary and composite documents. Specify `0` for the default flags.
insertCondTag	The condition tag to apply to insertions shown in the composite document. For no insert condition tag, specify an empty string.
deleteCondTag	The condition tag to apply to deletions shown in the composite document. For no insert condition tag, specify an empty string.
replaceText	Text to appear in place of the deleted text. For no replacement text, specify an empty string.
CompareThreshold	Percentage of words that can change before paragraphs are considered *not equal*. If two paragraphs are equal, word differences between them are shown within a paragraph in the composite document. If a paragraph is not equal to another, it is marked inserted or deleted. To specify an 85% threshold, set `compareThreshold` to `85`. The default value is `75`.

Details

This method is equivalent to using the Compare dialog box in FrameMaker. To make the various settings for the compare operation, you can logically OR the following values in the `Flags` argument:

flags constant	Meaning
`mFF_CMP_SUMMARY_ONLY`	Generate a summary document, but not a composite document
`mFF_CMP_CHANGE_BARS`	Turn on change bars in the composite document
`mFF_CMP_HYPERLINKS`	Put hypertext links in the summary document
`mFF_CMP_SUMKIT`	Open the summary document
`mFF_CMP_COMPKIT`	Open the composite document

Returns

A udtCompareRet data object. The members of this data object are compId and sumId—object IDs for the comparison document and summary document, respectively. If you are comparing books, compId is always `0`.

If Compare fails,it assigns of the following error codes to `FA_Errno`:

Error code	Meaning
`mFE_BadCompare`	olderId and newerId don't specify the same types of files.
`mFE_CompareTypes`	One of the files isn't a FrameMaker product document or book or one file is a book and the other is a document.
`mFE_WrongProduct`	Current FrameMaker product doesn't support the operation.

Examples

The following code opens two documents and compares them.

```
Dim oOldDoc As FMDocument, oNewDoc As FMDocument


Set oOldDoc = mySession.SimpleOpenDocument("C:\Temp\Test1.fm", 0)
Set oNewDoc = mySession.SimpleOpenDocument("C:\Temp\Test2.fm", 0)
oNewDoc.Compare oOldDoc, mFF_CMP_COMPKIT, "Inserted", "Deleted", "Replace", 75
```

Copy

Copies the current selection to the FrameMaker product Clipboard.

Synopsis

```
Method Synopsis.
```

Arguments

flags	Bit field that specifies how to copy the text and how to handle interactive alerts. For default settings, specify `0`.

Details

If you specify `0` for `flags`, `Copy` suppresses any interactive alerts or warnings that arise. You can also logically OR the following values into flags:

flags constant	Meaning
`mFF_INTERACTIVE`	Prompt user with dialog or alert boxes that arise
`mFF_STRIP_HYPERTEXT`	Do not copy any hypertext markers in the selection
`mFF_VISIBLE_ONLY`	Copy only the visible portion of the selection

The `mFF_INTERACTIVE` flag takes precedence over other flags. So, if you specify `mFF_INTERACTIVE | mFF_VISIBLE_ONLY` and the selection is not visible, the FrameMaker product allows the user to choose whether to copy the selection.

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFEWrongProduct`	Current FrameMaker product doesn't support operation
`mFEBadSelectionForOperation`	Selection doesn't support operation
`mFECanceled`	User or parameters canceled the operation
`mFEBadOperation`	Parameters specified an invalid operation

Examples

The following code replaces the selected range with the first 10 characters of the first paragraph in the text selection.

```
Dim oDoc As FMDocument, oPgf As FMParagraph
Dim f_tRange As udtTextRange, f_SaveRange As udtTextRange


Set oDoc = mySession.ActiveDoc
f_tRange = (oDoc.TextSelection)
f_SaveRange = f_tRange
Set oPgf = mySession.GetObject(f_tRange.beg.objId, oDoc)
f_tRange.beg.offset = 0
f_tRange.end.objId = oPgf.Id
f_tRange.end.offset = 10
```

```
oDoc.TextSelection = f_tRange
oDoc.Copy mFF_INTERACTIVE
oDoc.TextSelection = f_SaveRange
oDoc.Paste 0
```

CustomDoc

Creates a new custom document using the FrameMaker product's default new document template. For more information on default new document templates, see the FrameMaker documentation.

Synopsis

```
CustomDoc(width As Long,
height As Long,
numCols As Long,
columnGap As Long,
topMargin As Long,
botMargin As Long,
leftinsideMargin As Long,
rightoutsideMargin As Long,
sidedness As Long,
makeVisible As Long) As FMDocument
```

Arguments

Width	the width of the paper, expressed as F_Metric
height	The height of the paper, expressed as F_Metric
numCols	The number of columns for the text flow
columnGap	The gap between columns, expressed as F_Metric
topMargin	The top margin, expressed as F_Metric
botMargin	The bottom margin, expressed as F_Metric
leftInsideMargin	The left or inside margin, expressed as F_Metric
rightOutsideMargin	The right or outside margin, expressed as F_Metric
sidedness	A constant that specifies whether the document is single-sided or doublesided and on which side the document starts. Can be one of `mFF_Custom_SingleSided`, `mFF_Custom_FirstPageLeft`, or `mFF_Custom_FirstPageRight`

makeVisible	Whether or not to make the new document visible — `True` makes it visible

Details

Method description goes here.

Returns

The new FMDocument object. On success it sets `mFE_Success` to `FA_errno`, or one of the following error codes on failure:

Error code	Meaning
`FmE_BadParameter`	The function call specified an invalid parameter
`mFE_BadOperation`	The function call specified an illegal operation

Examples

Method Example.

Cut

Cuts the current selection to the FrameMaker product Clipboard.

Synopsis

```
Cut(flags As Long) As Long
```

Arguments

flags	Bit field that specifies how to cut the text and how to handle interactive alerts. For default settings, specify `0`.

Details

If you specify `0` for `flags`, `Cut` suppresses any interactive alerts or warnings that arise. You can also logically OR the following values into flags:

flags constant	Meaning
`mFF_INTERACTIVE`	Prompt user with dialog or alert boxes that arise
`mFF_CUT_TBL_CELLS`	Remove cut table cells
`mFF_VISIBLE_ONLY`	Cut only the visible portion of the selection
`mFF_DONT_DELETE_HIDDEN_TEXT`	Don't cut hidden conditional text

The `mFF_INTERACTIVE` flag takes precedence over other flags. So, if you specify `mFF_INTERACTIVE | mFF_DONT_DELETE_HIDDEN_TEXT` and the selection contains hidden text, the FrameMaker product allows the user to choose whether to cut the hidden text.

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFEWrongProduct`	Current FrameMaker product doesn't support operation
`mFEBadSelectionForOperation`	Selection doesn't support operation
`mFECanceled`	User or parameters canceled the operation
`mFEBadOperation`	Parameters specified an invalid operation

Examples

The following code cuts the first 10 characters from the first paragraph in the selection. It then replaces the selected range with those characters.

```
Dim oDoc As FMDocument, oPgf As FMParagraph
Dim f_tRange As udtTextRange, f_SaveRange As udtTextRange


Set oDoc = mySession.ActiveDoc
f_tRange = (oDoc.TextSelection)
f_SaveRange = f_tRange
Set oPgf = mySession.GetObject(f_tRange.beg.objId, oDoc)
f_tRange.beg.offset = 0
f_tRange.end.objId = oPgf.Id
f_tRange.end.offset = 10
oDoc.TextSelection = f_tRange
oDoc.Cut mFF_INTERACTIVE
oDoc.TextSelection = f_SaveRange
oDoc.Paste 0
```

DefineCommand

Defines an FMCommand object. After you define a command, you can add it to a menu with the AddCommandToMenu method.

Synopsis

```
DefineCommand(Id As Long,
name As String,
label As String,
shortcut As String) As FMCommand
```

Arguments

Id	The integer that the FrameMaker product passes to your plug-in's OnApiCommand event when the user executes the command. It must be unique for each menu item in your plug-in, but it need not be unique for different plug-ins.
name	A unique name for the command. If the user or a plug-in has already defined a command or menu with this name, the new command replaces it.
label	The title of the menu item as it appears on the menu.
shortcut.	The keyboard shortcut sequence. Many FrameMaker product commands use shortcuts beginning with Escape (\!).

Details

If you call `DefineCommand` and specify the name of a command that is already defined in the user's menu configuration files, the FrameMaker product gives precedence to the definition in the configuration files. If the configuration files assign a label or a shortcut to the command, the FrameMaker product uses it instead of the one you specify.

Returns

An FMCommand object. If `DefineCommand` fails, it assigns one of the following error codes to `mFA_Errno`:

Error code	Meaning
`mFE_WrongProduct`	Current FrameMaker product doesn't support this operation
`mFE_BadOperation`	Parameters specify an invalid operation
`mFE_BadParameter`	Parameter has an invalid value
`mFE_SystemError`	System error

Examples

The following code creates a menu and adds a command to it. The code then adds the new menu to the File menu.

```
Dim oMenu As FMMenu, oNewMenu As FMMenu
Dim oNewCommand As FMCommand

Set oNewMenu = mySession.DefineMenu("MyNewMenu", "My New Menu")
Set oNewCommand = mySession.DefineCommand(9902, "MyCommand", "MyCommand", "")
oNewCommand.AddCommandToMenu oNewMenu
```

```
Set oMenu = mySession.GetNamedObject("FileMenu", mFO_Menu, Nothing)
oMenu.AddMenuToMenu oNewMenu
```

DefineMenu

Defines an FMMenu object. After you define a command, you can add it to a menu or menu bar with the AddMenuToMenu method.

Synopsis

```
DefineMenu(name As String,
label As String) As FMMenu
```

Arguments

name — A unique name for the menu. If the user or a plug-in has already defined a command or menu with this name, the new menu replaces it.

label — The title of the menu as it appears in the FrameMaker product.

Details

If you call `DefineMenu` and specify the name of a menu that is already defined in the user's menu configuration files, the FrameMaker product gives precedence to the definition in the configuration files. If the configuration files assign a label or a shortcut to the command, the FrameMaker product uses it instead of the one you specify.

Returns

An FMMenu object. If `DefineCommand` fails, it assigns one of the following error codes to `mFA_Errno`:

Error code	Meaning
`mFE_WrongProduct`	Current FrameMaker product doesn't support this operation
`mFE_BadOperation`	Parameters specify an invalid operation
`mFE_BadParameter`	Parameter has an invalid value
`mFE_SystemError`	System error

Examples

The following code creates a menu and adds a command to it. The code then adds the new menu to the File menu.

```
Dim oMenu As FMMenu, oNewMenu As FMMenu
Dim oNewCommand As FMCommand


Set oNewMenu = mySession.DefineMenu("MyNewMenu", "My New Menu")
```

```
Set oNewCommand = mySession.DefineCommand(9902, "MyCommand", "MyCommand", "")
oNewCommand.AddCommandToMenu oNewMenu

Set oMenu = mySession.GetNamedObject("FileMenu", mFO_Menu, Nothing)
oMenu.AddMenuToMenu oNewMenu
```

Delete

Deletes an object from a document. Be careful about possible side effects when you use `Delete`. When you delete some objects, FrameAC deletes any objects they contain. For example, when you delete an FMTable, FrameAC deletes all the FMRow and FMCell objects that it contains. When you delete an FMAnchoredFrame, FrameAC deletes all the objects in it.

Synopsis

```
Delete() As Long
```

Details

The following objects can't be deleted via the Delete method, even though they have Delete as a member:

- FMBook
- FMDocument
- FMHiddenPage
- FMCell
- FMRow
- FMMasterPage (Left and Right master pages, only)
- FMPageFrame
- FMColor
- FMSession
- FMVariableFormat (system variables, only)

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_BadDelete`	Specified object couldn't be deleted
`mFE_BadOperation`	Function call specified an illegal operation
`mFE_BadParameter`	Function call specified an invalid parameter

Examples

The following code deletes all the markers in a document.

```
Dim oDoc As FMDocument, oMarker As FMMarker, oTmp As FMMarker


Set oDoc = mySession.ActiveDoc
Set oMarker = oDoc.FirstMarkerInDoc
Do Until oMarker Is Nothing
     Set oTmp = oMarker
     Set oMarker = oTmp.NextMarkerInDoc
     oTmp.Delete
Loop
```

DeleteBodyRows

Deletes all the body rows from a table, leaving a single, empty body row.

Synopsis

```
DeleteBodyRows() As Long
```

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_WrongProduct`	Current FrameMaker product doesn't support the requested operation.
`mFE_BadOperation`	The operation cannot be performed.

Examples

The following code deletes body rows from the current table.

```
Dim oDoc As FMDocument, oTbl As FMTable


Set oDoc = mySession.ActiveDoc
Set oTbl = oDoc.SelectedTbl
oTbl.DeleteBodyRows
```

DeleteCellText

Clears the text out of the FMCell object.

Synopsis

```
DeleteCellText() As Long
```

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_WrongProduct`	Current FrameMaker product doesn't support the requested operation.
`mFE_BadOperation`	The operation cannot be performed.

Examples

The following code deletes the text from the first column of the current table.

```
Dim oDoc As FMDocument, oTbl As FMTable, oRow As FMRow, oCell As FMCell


Set oDoc = mySession.ActiveDoc
Set oTbl = oDoc.SelectedTbl
Set oRow = oTbl.GetFirstRow(mFV_ROW_BODY)
Do Until oRow Is Nothing
     Set oCell = oRow.FirstCellInRow
     oCell.DeleteCellText
     Set oRow = oRow.NextRowInTbl
Loop
```

DeleteCols

Deletes columns from a table. To delete an entire table, use Delete.

Synopsis

```
DeleteCols(FirstColToDelete As Long,
Count As Long) As Long
```

Arguments

`FirstColToDelete` The first column to delete—columns are numbered from left to right, starting from 0.

`Count` The number of columns to delete.

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_WrongProduct	Current FrameMaker product doesn't support the requested operation.
mFE_BadOperation	The operation cannot be performed.
mFE_TableInLockedTi	The table is in a locked text inset

Examples

The following code deletes the second column in the current table.

```
Dim oDoc As FMDocument, oTbl As FMTable


Set oDoc = mySession.ActiveDoc
Set oTbl = oDoc.SelectedTbl
oTbl.DeleteCols 1, 1
```

DeleteRowText

Clears the text out of the FMRow object.

Synopsis

```
DeleteRowText() As Long
```

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_WrongProduct	Current FrameMaker product doesn't support the requested operation.
mFE_BadOperation	The operation cannot be performed.

Examples

The following code deletes the text from every even row of the current table.

```
Dim oDoc As FMDocument, oTbl As FMTable, oRow As FMRow
Dim count As Integer


Set oDoc = mySession.ActiveDoc
Set oTbl = oDoc.SelectedTbl
Set oRow = oTbl.GetFirstRow(mFV_ROW_BODY)
```

```
count = 1
Do Until oRow Is Nothing
     If count Mod 2 Then
          oRow.DeleteRowText
     End If
     Set oRow = oRow.NextRowInTbl
     count = count + 1
Loop
```

DeleteRows

Deletes rows from a table. Like the Delete command in the FrameMaker product user interface, DeleteRows does not allow you to delete more than one type of row at time. The range of rows you specify must be all body rows, all header rows, or all footer rows.

Synopsis

```
DeleteRows(FirstRowToDelete As FMRow,
Count As Long) As Long
```

Arguments

FirstRowToDelete The first FMRow object to delete.

Count The number of rows to delete.

Details

Method description goes here.

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_WrongProduct	Current FrameMaker product doesn't support the requested operation.
mFE_BadOperation	The operation cannot be performed.
mFE_OutOfRange	The specified range includes more than one type of row (for example, header rows and body rows)

Examples

The following code deletes every odd body row from the table.

```
Dim oDoc As FMDocument, oTbl As FMTable, oRow As FMRow, oTmp As FMRow
Dim count As Integer
```

```
Set oDoc = mySession.ActiveDoc
Set oTbl = oDoc.SelectedTbl
Set oRow = oTbl.GetFirstRow(mFV_ROW_BODY)
count = 1
Do Until oRow Is Nothing
     Set oTmp = oRow
     Set oRow = oRow.NextRowInTbl
     If count Mod 2 Then
          oTbl.DeleteRows oTmp, 1
     End If
     count = count + 1
Loop
```

DeleteText

Deletes a specified text range from a document.

Synopsis

```
DeleteText(textRange) As Long
```

Arguments

textRange. The range of text to delete.

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_BadDelete`	Specified text couldn't be deleted
`mFE_BadObjId`	Invalid object ID
`mFE_BadRange`	Specified text range is invalid
`mFE_NotTextObject`	Object specified for the text range isn't an object that contains text, for example, a text frame (FO_TextFrame), a paragraph (FO_Pgf) or a text line (FO_TextLine)
`mFE_OffsetNotFound`	Offset specified for the text range couldn't be found in the specified paragraph or text line
`mFE_BadSelectionForOperation`	Selection is within a locked text range.

Examples

The following code deletes text to the left of the current selection's start location.

```
Dim oPgf As FMParagraph
Dim oDoc As FMDocument
Dim f_tRange As udtTextRange

Set oDoc = mySession.ActiveDoc
f_tRange = oDoc.TextSelection
Set oPgf = mySession.GetObject(f_tRange.beg.objId, oDoc)

'Now extend the range to the left of the first position.
f_tRange.end.objId = oPgf.Id
f_tRange.end.offset = f_tRange.beg.offset
f_tRange.beg.offset = 0
oDoc.DeleteText f_tRange
```

DeleteTextInsetContents

Deletes the text in a text inset. You must unlock the text inset before you call this method to delete its contents. After you are done, you must relock the text inset.

Synopsis

```
DeleteTextInsetContents() As Long
```

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFEBadDelete`	Specified text couldn't be deleted
`mFEBadSelectionForOperation`	The current text inset is locked

Examples

The following code deletes the content from the first text inset in the current document.

```
Dim oDoc As FMDocument
Dim oTextInset As FMTextInset

Set oDoc = mySession.ActiveDoc
Set oTextInset = oDoc.FirstTiInDoc
oTextInset.TiLocked = False
oTextInset.DeleteTextInsetContents
oTextInset.TiLocked = True
```

DeleteUndefinedAttribute

Deletes attribute for which no value is assigned. You can delete undefined attributes for a given FMElement object, FMElementDefinition object (all instances of elements that use this definition) , or FMDocument object (all elements in the document).

Synopsis

```
DeleteUndefinedAttribute(AttributeName As String) As Long
```

Arguments

AttributeName The name of the attribute you want to delete.

Returns

mFE_Success if it succeeds.

Examples

The following code loops through the children of the highest-level element. For each element with an undefined attribute named "Deprecated", the code deletes the attribute.

```
Dim oDoc As FMDocument
Dim oFlow As FMFlow
Dim oElem As FMElement

Set oDoc = mySession.ActiveDoc
Set oFlow = oDoc.MainFlowInDoc
Set oElem = oFlow.HighestLevelElement.FirstChildElement
Do Until oElem Is Nothing
     oElem.DeleteUndefinedAttribute "Deprecated"
     mySession.CallClient "cudLog", "logMessage@" + oElem.ElementDef.Name
     Set oElem = oElem.NextSiblingElement
Loop
```

DemoteElement

Demotes the selected structural element or elements. The element becomes a child of the sibling element before it.

Synopsis

```
DemoteElement() As Long
```

Details

This is a method of the current FMDocument object. To use this method, you must have at least one structural element completely selected.

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_BadSelectionForOperation	Current text selection is invalid for this operation

Examples

The following code demotes the currently selected element.

```
Set oDoc = mySession.ActiveDoc
oDoc.DemoteElement
```

ElementLocToTextLoc

Returns a text location that corresponds with the current element location.

Synopsis

```
ElementLocToTextLoc(ElementLoc)
```

Arguments

ElementLoc. The udtElementLoc to convert

Returns

The udtTextLoc data specifying a text location. If ElementLocToTextLoc fails, it sets the following value to mFA_Errno.

Error code	Meaning
mFE_BadParameter	ElementLoc was empty, or one of its values was improperly specified

Examples

The following code sets the insertion point to be at the end location of the current element range.

```
Dim f_tLoc As udtTextLoc, f_tRange As udtTextRange
Dim f_elemLoc As udtElementLoc, f_elemRange As udtElementRange


Set oDoc = mySession.ActiveDoc
f_elemRange = oDoc.ElementSelection
f_tLoc = oDoc.ElementLocToTextLoc(f_elemRange.end)
f_tRange.beg = f_tLoc
f_tRange.end = f_tLoc
oDoc.TextSelection = f_tRange
```

ExecuteScript

Executes a FrameAC script.

Synopsis

```
ExecuteScript(Script As String,
ScriptType As AScriptType) As Long
```

Arguments

Script	The full path to the script file
ScriptType	The type of script to execute. Can be one of: ast_FILE ast_TEXT_JSCRIPT ast_TEXT_VBSCRIPT

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_BadParameter`	The function call specified an invalid parameter
`mFE_BadOperation`	The function call specified an illegal operation

Examples

The following code executes a script that was saved to disk.

```
mySession.ExecuteScriptWithParam "C:\MyWork\MyScripts\Tutoria1.vbs",
     ast_TEXT_VBSCRIPT
```

ExecuteScriptWithParam

Executes a FrameAC script and allows you to pass a class as a parameter to the script. The script can set / use the class members and the changed member values (if any) will be reflected in VisualBasic.

Synopsis

```
ExecuteScriptWithScript(Script As String,
ScriptType As AScriptType, Param as Object) As Long
```

Arguments

Script	The full path to the script file
ScriptType	The type of script to execute. Can be one of: ast_FILE ast_TEXT_JSCRIPT ast_TEXT_VBSCRIPT

Param	The class you would like passed.

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_BadParameter`	The function call specified an invalid parameter
`mFE_BadOperation`	The function call specified an illegal operation

Examples

The following code executes a script that was saved to disk with a passed class whose values can be set in the script and returned to the VB code.

```
mySession.ExecuteScriptwithParam "C:\MyWork\MyScripts\Tutoria1.vbs",
     ast_TEXT_VBSCRIPT, clsData
```

Find

Performs the same actions as using the Find dialog box to search a document for text or other types of content.

Synopsis

```
Find(textloc, FindParameters)
```

Arguments

textLoc	The udtTextLoc that describes where to begin searching
FindParameters	An array of properties that specifies what to search for

Details

The findParamsp parameter points to a property list that contains:

- `mFS_FindCustomizationFlags`, an optional property you can use to customize the search.
- A list of a list of properties that specify what type of content to search for; text, elements, character formats, etc. You must specify one of these properties.

You can assign the following properties to FindParameters. Depending on the type of search you want to perform, you must assign a subset of these listed properties and values. For example, if you are searching for text, then you should not assign any strings to `mFS_FindElementTag`. However, you must assign a string to `mFS_FindText`.

Property	Meaning and possible values
mFS_FindCustomizationFlags	An optional parameter of type mFT_Integer that may be any of the following bit flags OR'ed together: mFF_FIND_CONSIDER_CASE, mFF_FIND_WHOLE_WORD, mFF_FIND_USE_WILDCARDS, mFF_FIND_BACKWARDS If no customization flags are specified the default is to search forward, to not use wildcards, to not consider case, and to not use whole words.
mFS_FindWrap	A Boolean value that determines whether the find operation will wrap when it reaches the location where the search began. Default is True; the find operations wraps. If False, after reaching the location where the search began, the find operation returns an empty udtTextRange and FA_errno is set to mFE_NotFound.
mFS_FindText	The string of text to search for.
mFS_FindElementTag	A list of strings, as follows: List length = mFV_NumFindElementItems; List element mFV_FindElemTag = an element tag name; List element mFV_FindAttrName = an attribute name List element mFV_FindAttrValue = an attribute value All of the strings must be present, but any or all may be empty

Property	Meaning and possible values
mFS_FindCharFmt	You do not assign values to this property. Instead, you assign one or more of the following properties to FindParameters to describe the character formatting you want to find: mFP_FontFamily mFP_CombinedFont mFP_FontSize mFP_FontAngle mFP_FontWeight mFP_FontVariation mFP_Color mFP_Spread mFP_Stretch mFP_Language mFP_Underline mFP_Overline mFP_Strikethrough mFP_ChangeBar mFP_Capitalization mFP_Position mFP_Tsume
mFS_FindPgfTag	A string for the paragraph tag
mFS_FindCharTag	A string for the character tag
mFS_FindTableTag	A string for the table tag

Property	Meaning and possible values
mFS_FindObject	An integer indicating the object type: mFV_FindAnyMarker mFV_FindAnyXRef mFV_FindUnresolvedXRef mFV_FindAnyTextInset mFV_FindUnresolvedTextInset mFV_FindAnyPub mFV_FindAnyVariable mFV_FindAnchoredFrame mFV_FindFootnote mFV_FindAnyTable mFV_FindAutomaticHyphen mFV_FindAnyRubi
mFS_FindMarkerOfType	A string indicating the marker type
mFS_FindMarkerText	A string for the marker text
mFS_FindXRefWithFormat	A string for the cross-reference format
mFS_FindNamedVariable	A string for the variable name
mFS_FindCondTextInCondTags	A list of strings for the condition tags
mFS_FindCondTextNotInCondTags	A list of strings for the condition tags

Whenever this function finds something that corresponds to a text range (a word, object anchor, marker, etc.), it returns udtTextRange data for that range. However, when searching for structure elements, you can find elements that have no corresponding text range.

Structure elements for the following table parts have no corresponding text range:

- Table title
- Table head
- Table foot
- Table body
- Table row
- Table cell

When you find a structure element for one of these objects, the Find method returns empty udtTextRange data and FA_errno is set to mFE_Success. In this case, you can get the FMDocument object's ElementSelection property to return a corresponding udtElementRange for the table part structure element.

Returns

When it finds anything other than structure elements for table parts, this method returns udtTextRange data. When it finds structure elements for table parts, this method returns empty udtTextRange data, and `FA_errno` is set to `FE_Success`.

If Find fails, it returns empty udtTextRange data, and one of the following error codes is set to `FA_errno`:

Error code	Meaning
`mFE_BadParameter`	`FindParams` was empty or a parameter was improperly specified
`mFE_BadInsertPos`	The passed text location was not valid
`mFE_NotTextObject`	The passed text location was notactually a text location

Examples

The following code searches the current document for a string.

```
Dim oDoc As FMDocument
Dim f_tLoc As udtTextLoc, f_tRange As udtTextRange
Dim findParams(0) As Variant, findProps As udtPropVal
Dim text As String

'Set up the findProps
text = "search for this string"
findProps.propIdent.num = mFS_FindText
findProps.propVal.valType = mFT_String
findProps.propVal.Value = text
findParams(0) = findProps

' Perform the find, then center the found text in the doc window.
Set oDoc = mySession.ActiveDoc
f_tRange = oDoc.TextSelection
f_tRange = oDoc.Find(f_tRange.beg, findParams)
oDoc.CenterOnText f_tRange
```

Generate

Generates files for a book. It performs the same operation as choosing Update Book from the book Edit menu. The book and its generated files must be set up before you call this method.

Synopsis

```
Generate(interactive As Long,
makeVisible As Long) As Long
```

Arguments

Interactive	Specifies whether to display warnings, messages, and the book error log to the user. True dsplays messages and warnings.
makeVisible	Specifies whether to display the generated files. True displays the files.

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_BadParameter`	The function call specified an invalid parameter
`mFE_BadOperation`	The book is not self-consistent (book generates data in one file that is source data for another generated file, or page count continually changes for this operation); there is a duplicate file in the book; all files in the book are generated files
`mFE_SystemError`	Couldn't allocate memory, or couldn't open or save one of the files in the book

Examples

The following code generates files for a book.

```
Dim oBook As FMBook


Set oBook = mySession.ActiveBook
oBook.Generate False, True
```

GetCountOfRowsOfType

Returns the number of rows of the specified type that are in the FMTable object..

Synopsis

```
GetCountOfRowsOfType(RowType As Long) As Long
```

Arguments

RowType	The type of row to count. Can be one of:
	`mFV_ROW_BODY`
	`mFV_ROW_HEADING`
	`mFV_ROW_FOOTING`

Returns

The number of rows of the specified type that are in the table.

Examples

The following code deletes all but the last row in the current table.

```
Dim oTbl As FMTable, oRow As FMRow, oTmp As FMRow
Dim oDoc As FMDocument
Dim rowCount As Long, i As Long

Set oDoc = mySession.ActiveDoc
Set oTbl = oDoc.SelectedTbl
rowCount = oTbl.GetCountOfRowsOfType(mFV_ROW_BODY)
Set oRow = oTbl.GetFirstRow(mFV_ROW_BODY)
For i = 2 To rowCount
     Set oTmp = oRow
     Set oRow = oRow.NextRowInTbl
     oTbl.DeleteRows oTmp, 1
Next i
```

GetElementCatalog

Queries the FMDocument object for the list of element definitions that appear in the element catalog for the current position in the structured FrameMaker document.

Synopsis

```
GetElementCatalog()
```

Details

The returned list of element definitions matches the list displayed in the Element Catalog on the user's screen. Its contents depend on the value of the current document's `mFP_ElementCatalogDisplay` property. For example, if the document's `mFP_ElementCatalogDisplay` property is set to `mFV_ELCAT_CHILDREN`, the list returned by `GetElementCatalog` contains children allowed anywhere in the current parent element.

Returns

An array of FMElementDefinition objects on success. On success it sets `mFE_Success` to FA_errno, or one of the following error codes on failure:

Error code	Meaning
`FE_BadParameter`	The function call specified an invalid parameter

Error code	Meaning
FE_BadOperation	The function call specified an illegal operation

Examples

Method Example.

GetFirstRow

Gets the first row of a table's heading, body, or footing.

Synopsis

```
GetFirstRow(RowType As Long) As FMRow
```

Arguments

RowType — The type of row to get. Can be one of:

mFV_ROW_BODY

mFV_ROW_HEADING

mFV_ROW_FOOTING

Returns

The first row of the specified type that is in the table.

If the method fails, it sets one of the following error codes to FA_errno:

Error code	Meaning
FE_BadParameter	The function call specified an invalid parameter
FE_BadOperation	The function call specified an illegal operation

Examples

The following code gets the first body row in the current table.

```
Dim oDoc As FMDocument, oRow As FMRow


Set oDoc = mySession.ActiveDoc
Set oRow = oDoc.SelectedTbl.GetFirstRow(mFV_ROW_BODY)
```

GetImportDefaultParams

Returns a default property list that you can use to with the Import method.

Synopsis

```
GetImportDefaultParams()
```

Details

This method returns a list of properties that you can use to script the behavior of the Import method. By changing the settings of these properties, you can determine what type of file to import and what to do under different circumstances. The following tables list the parameters, showing the default value for each, followed by the other values you can set.

The returned property list contains all the properties shown in these tables. However, the Import method uses some of these properties only for certain types of import operations. For example, it uses mFS_UseMainFlow only when importing a FrameMaker document or MIF file — it ignores this property when importing text or graphic files.

Properties for All Import Operations:

Property	Instruction or situation and possible values
`mFS_AlertUserAboutFailure`	Alert user if an unexpected condition, such as an unrecognized file type, occurs. `False`: don't notify user when unexpected conditions occur. `True`: notify user when unexpected conditions occur.
`mFS_DisallowDoc`	Disallow Frame binary documents. `False`: allow them to be imported. `True`: don't allow them to be imported.
`mFS_DisallowFilterTypes`	Disallow filterable files. `False`: allow them to be imported. `True`: don't allow them to be imported.
`mFS_DisallowMIF`	Disallow MIF files. `False`: allow them to be imported. `True`: don't allow them to be imported.
`mFS_DisallowGraphicTypes`	Disallow graphic files. `False`: allow them to be imported. `True`: don't allow them to be imported.
`mFS_DisallowMacEditions`	Disallow Macintosh editions. `False`: allow them to be imported. `True`: don't allow them to be imported.
`mFS_DisallowPlainText`	Disallow Text Only files. `False`: allow them to be imported. `True`: don't allow them to be imported.

Property	Instruction or situation and possible values
mFS_DisallowSgml	Disallow SGML documents. False: allow them to be imported. True: don't allow them to be imported.
mFS_DisallowXml	Disallow XML documents. False: allow them to be imported. True: don't allow them to be imported.
mFS_DontNotifyAPIClients	Notify other FrameAC plugins and FDK clients of the import operation. True: don't notify them. False: notify them.
mFS_FileTypeHint	If the file is filterable, a string that enables the FrameMaker product to automatically call the correct filter to filter it. For information on the syntax of this string, see "Graphic and Text Inset Hint Strings" on page 407. Empty string.
mFS_FileIsSgmlDoc	File is an SGML document. mFVDoOK: import it anyway. mFVDoCancel: cancel import operation. mFVDoShowDialog: show dialog box and let user decide.
mFS_FileIsXmlDoc	File is an XML document. mFVDoOK: import it anyway. mFVDoCancel: cancel import operation. mFVDoShowDialog: show dialog box and let user decide.
mFS_ForceImportAsText	Import file as a Text Only document, even if it is a MIF file or a filterable file. Note that some files, such as FrameMaker binary files, cannot be imported as text. False: import it in a format based on its type True: import it as Text Only.
mFS_HowToImport	Import file by reference or copy. mFVDoByRef: import file by reference. mFVDoByCopy: import file by copy. mFVDoUserChoice: allow user to choose how to import the file.

Property	Instruction or situation and possible values
mFS_ImportAsType	Specify the format of the file to import. mFVAUTORECOGNIZE: Default value; recognize the file type aoutmatically. mFVTYPE_BINARY: A FrameMaker binary file. mFVTYPE_MIF mFVTYPE_TEXT mFVTYPE_SGML mFVTYPE_XML mFVTYPE_FILTER: Use a filter to import this file. You must specify a valid file type hint for mFS_FileTypeHint.
mFS_ManualUpdate	Update inset manually. False: don't update inset manually. True: update inset manually.
mFS_SgmlImportApplication	String specifying the name of the structure application to use when importing an XML or SGML file. This paramater takes precedence over any other structure application specification. If the specified application doesn't exist, the calling method will fail. Empty string (No Application Used)
mFS_ShowBrowser	Display Import dialog box. False: don't display it. True: display it.
mFS_TextInsetName	Inset name. Empty string

Properties only for importing FrameMaker binary and MIF files:

Property	Instruction or situation and possible values
mFS_FileIsMakerDoc	File is a Frame binary document or a MIF file mFVDoOK: import it anyway mFVDoCancel: cancel import operation mFVDoShowDialog: show dialog box and let user decide

Property	Instruction or situation and possible values
mFS_FormatImportedText	Format the imported text
	mFVEnclosingDoc: use formatting in the enclosing document
	mFVPlainText: format the imported text as plain text
	mFVSourceDoc: use formatting from the source document
mFS_ImportFlowPageSpace	If mFS_UseMainFlow is False, the type of pages to search for the flow specified by mFS_ImportFlowTag
	mFVBodyPage: search body pages
	mFVReferencePage: search reference pages
mFS_ImportFlowTag	If mFS_UseMainFlow is False, the name of the flow to import
	Empty string
mFS_RemoveManualPageBreaks	Remove manual page breaks if mFS_FormatImportedTest is set to mFVEnclosingDoc
	True: remove manual page breaks
	False: don't remove manual page breaks
mFS_RemoveOverrides	Remove format overrides if mFS_FormatImportedTest is set to mFVEnclosingDoc
	True: remove format overrides
	False: don't remove format overrides
mFS_UseMainFlow	Import text from specified document's main flow
	True: import the text from the main flow
	False: don't import the text from the main flow

Properties only for importing Graphics files:

Property	Instruction or situation and possible values
mFS_FileIsGraphic	File is a graphic file
	mFVDoOK: import it
	mFVDoCancel: cancel import operation
	mFVDoShowDialog: display dialog box and let user decide
mFS_FitGraphicInSelectedRect	Fit the graphic in the selected graphic frame
	True: fit the graphic in the frame
	False: don't fit the graphic in the frame
mFS_GraphicDpi	Integer specifying dots per inch (DPI) at which to import the graphic
	72 (to specify 72 dpi)

Properties only for importing ASCII text files:

Property	Instruction or situation and possible values
mFS_CellSeparator	If mFS_FileIsText is FV_DoImportAsTable, the delimiter or separator used to parse the text into cells
	Empty string
mFS_FileIsText	File is a Text Only file
	FV_TextFile_EOLisEOP: import it and convert each end-of-line into a paragraph break
	FV_TextFile_EOLisNotEOP: import it but don't convert each end-of-line into a paragraph break
	FV_DoImportAsTable: import it into a table.
	FV_DoCancel: cancel Import operation
mFS_ImportTblTag	If mFS_FileIsText is FV_DoImportAsTable, the table format to use
	Empty string
mFS_LeaveHeadingRowsEmpty	If mFS_FileIsText is FV_DoImportAsTable, leave heading rows empty
	False: don't leave heading rows empty
	True: leave heading rows empty
mFS_NumCellSeparators	If mFS_FileIsText is FV_DoImportAsTable, and mFS_CellSeparator is set to a space (' '), the number of spaces to use as a separator
	1
mFS_NumColumns	If mFS_FileIsText is FV_DoImportAsTable, and mFS_TreatParaAsRow is False, the number of columns in the table
	1
mFS_TblNumHeadingRows	If mFS_FileIsText is FV_DoImportAsTable, the number of heading rows in the table
	1
mFS_TreatParaAsRow	If mFS_FileIsText is FV_DoImportAsTable, convert each line in the text file into a row of table cells and use mFS_CellSeparator and mFS_NumCellSeparators to determine how to divide the line into separate cells
	True: convert each line into a row of table cells
	False: convert each line into a table cell instead

When you import text into a table, in addition to setting mFS_FileIsText to mFV_DoImportAsTable, you must specify a value for the mFS_ImportTblTag property. If you set the property mFS_TreatParaAsRow to True, you must also specify a value for the mFS_CellSeparator property.

The property list returned by GetImportDefaultParams does not specify values for the mFS_ImportTblTag and mFS_CellSeparator properties. If you use the property list to import a table and do not specify a value for mFS_ImportTblTag, the Import method will fail and set FA_errno to mFE_BadParameter. If you set mFS_TreatParaAsRow to True and do not specify a cell separator by setting mFS_CellSeparator, the Import method will fail and set FA_errno to mFE_BadParameter.

Returns

A list of import properties. If it fails, the list is empty.

Examples

See the example for the Import method.

GetLastRow

Gets the last row of a table's heading, body, or footing.

Synopsis

```
GetLastRow(RowType As Long) As FMRow
```

Arguments

RowType	The type of row to get. Can be one of: mFV_ROW_BODY mFV_ROW_HEADING mFV_ROW_FOOTING

Returns

The last row of the specified type that is in the table.

If the method fails, it sets one of the following error codes to FA_errno:

Error code	Meaning
FE_BadParameter	The function call specified an invalid parameter
FE_BadOperation	The function call specified an illegal operation

Examples

The following code gets the last body row in the current table.

```
Dim oDoc As FMDocument, oRow As FMRow
```

```
Set oDoc = mySession.ActiveDoc
Set oRow = oDoc.SelectedTbl.GetLastRow(mFV_ROW_BODY)
```

GetNamedObject

Gets an object of a specific name and object type from the specified book or document.

Synopsis

```
GetNamedObject(ObjectName As String,
ObjectType As FDK_ObjectDefs,
DocumentOrBook As FMObject) As FMObject
```

Arguments

ObjectName	The name of the object you want to get.
ObjectType	The type of object to get (for example, `mFO_TblFmt`).
DocumentOrBook	The FMBook or FMDocument object that contains the named object you want to get.

Details

You can use this method to get objects of the following types:

- mFO_CharFmt
- mFO_Color
- mFO_CombinedFontDefn
- mFO_Command
- mFO_CondFmt
- mFO_ElementDef
- mFO_FmtChangeList
- mFO_Menu
- mFO_MenuItemSeparator
- mFO_MasterPage
- mFO_PgfFmt
- mFO_RefPage
- mFO_RulingFmt
- mFO_TblFmt
- mFO_UnanchoredFrame (reference frame)

- mFO_VarFmt
- mFO_XRefFmt

You can see a list of all object types in the Visual Basic environment's Object Browser under FDK_ObjectDefs.

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_TypeUnNamed`	Objects of the specified type do not have names
`mFE_NameNotFound`	Object with the specified name and type doesn't exist in the specified document

Examples

The following code gets the paragraph format named "MyFmt" and deletes it from the document.

```
Dim oDoc As FMDocument, oPgfFmt As FMParagraphFormat


Set oDoc = mySession.ActiveDoc
Set oPgfFmt = mySession.GetNamedObject("button", mFO_PgfFmt, oDoc)
oPgfFmt.Delete
```

GetObject

Gets the object that corresponds to a specified object ID.

Synopsis

```
GetObject(ObjectId As Long,
DocumentOrBook As FMObject) As FMObject
```

Arguments

`ObjectId` The ID of the object you want to get.

`DocumentOrBook.` The FMBook or FMDocument object that contains the named object you want to get.

Details

Every object has an ID that identifies it within a given document, session, or book. This ID is a Long number that is unique for the given object. However, the ID only persists as long as the document or book is open — when you close then open the file that contains the object, FrameMaker assigns it a new ID.

In most cases you can get an object directly as properties of their parent objects. For example, any FMDocument object has a first FMFlow object. From that flow object, you can get loop for the next FMFlow in the document checking for flow names or other criteria.

There are some data types that pertain to objects, but refer to them via object ID. In order to operate on these objects, you must convert the ID to an actual object. For example, a text location is expressed as an object ID for a paragraph or text line, and an offset from the beginning of that object. If you want to operate on the paragraph object, you must use the GetObject method to convert the ID into a paragraph object.

Returns

The associated object if it succeeds, or Nothing on failure.

Examples

The following code gets the paragraph that contains the beginning of the text selection in the active document.

```
Dim f_tRange As udtTextRange
Dim oDoc As FMDocument, oPgf As FMParagraph


Set oDoc = mySession.ActiveDoc
f_tRange = oDoc.TextSelection
Set oPgf = mySession.GetObject(f_tRange.beg.objId, oDoc)
```

GetOpenDefaultParams

Returns a default property list that you can use to with the OpenBook or OpenDocument method.

Synopsis

```
GetOpenDefaultParams()
```

Details

This method returns a list of properties that you can use to script the behavior of the OpenBook or OpenDocument method. By changing the settings of these properties, you can determine what type of file to open and what to do under different circumstances. The following tables list the parameters, showing the default value for each, followed by the other values you can set.

Properties for All Open Operations:

Property	Instruction or situation and possible values
mFS_AlertUserAboutFailure	Alert user if an unexpected condition, such as an unrecognized file type, occurs. False: don't notify user when unexpected conditions occur. True: notify user when unexpected conditions occur.
mFS_BookIsInUse	Book is already open. mFV_OpenViewOnly: open a view-only copy of the book. mFV_OpenEditableCopy: open an editable copy of the book. mFV_DoShowDialog: show dialog box and let user decide. mFV_ResetLockAndContinue: reset lock and open file. mFV_DoCancel: cancel Open operation.
mFS_DisallowBookDoc	File is a FrameMaker binary book. False: open it anyway. True: don't open it.
mFS_DisallowBookMIF	File is a MIF book. False: open it anyway. True: don't open it.
mFS_DisallowDoc	File is a FrameMaker binary document. False: open it anyway. True: don't open it.
mFS_DisallowFilterTypes	Disallow filterable files. False: open it anyway. True: don't open it.
mFS_DisallowMIF	File is a MIF document. False: open it anyway. True: don't open it.
mFS_DisallowPlainText	File is Text Only. False: open it and conver it. True: don't open it.
mFS_DisallowSgml	File is an SGML document. False: open it and conver it. True: don't open it.

Property	Instruction or situation and possible values
mFS_DisallowXml	File is an XML document. False: open it and convert it. True: don't open it.
mFS_DontNotifyAPIClients	Don't notify other FrameAC plugins and FDK clients of the import operation. False: notify them. True: don't notify them.
mFS_FileIsInUse	File is already open. mFV_OpenViewOnly: open in View Only format. mFV_OpenEditableCopy: open an editable copy. mFV_DoShowDialog: show dialog box and let user decide. mFV_ResetLockAndContinue: reset the lock and open the file. mFV_DoCancel: cancel Open operation.
mFS_FileIsOldVersion	File is from previous version of the FrameMaker product. mFV_DoCancel: cancel Open operation. mFV_DoOK: open it anyway. mFV_DoShowDialog: show dialog box and let user decide.
mFS_FileIsStructured	File has Structured FrameMaker features, but current product interface isn't Structured FrameMaker. mFV_OpenViewOnly: open a View Only copy of file. mFV_DoCancel: cancel Open operation. mFV_StripStructureAndOpen: remove structure features and open file. mFV_DoShowDialog: show dialog box and let user decide.
mFS_FileIsText	File is Text Only. mFV_TextFile_EOLisEOP: open it and convert each end-of-line into a paragraph break. mFV_TextFile_EOLisNotEOP: open it but don't convert each end-of-line into a paragraph break. mFV_DoShowDialog: show dialog box and let user decide. mFV_DoCancel: cancel Open operation.

Property	Instruction or situation and possible values
mFS_FileTypeHint	If the file is filterable, a string that enables the FrameMaker product to automatically call the correct filter to filter it. For information on the syntax of this string, see "Graphic and Text Inset Hint Strings" on page 407.
	Empty string.
mFS_FontChangedMetric	A font metric needs to be changed.
	mFV_DoCancel: cancel Open operation.
	mFV_DoOK: open the document anyway.
	mFV_DoShowDialog: show dialog box and let user decide.
mFS_FontNotFoundInCatalog	Catalog contains fonts that aren't available.
	mFV_DoCancel: cancel Open operation.
	mFV_DoOK: open it anyway.
	mFV_DoShowDialog: show dialog box and let user decide.
mFS_FontNotFoundInDoc	Document uses unavailable fonts.
	mFV_DoCancel: cancel Open operation.
	mFV_DoOK: open it anyway.
	mFV_DoShowDialog: show dialog box and let user decide.
mFS_ForceImportAsText	Open file as a Text Only document, even if it is a MIF file or a filterable file. Note that some files, such as FrameMaker binary files, cannot be opened as text.
	False: open it in a format based on its type
	True: open it as Text Only.
mFS_LanguageNotAvailable	The file uses an unavailable language.
	mFV_DoCancel: cancel Open operation.
	mFV_DoOK: open it anyway.
	mFV_DoShowDialog: show dialog box and let user decide.
mFS_LockCantBeReset	Attempted to reset FrameMaker product file lock but wasn't able to.
	mFV_DoCancel: cancel Open operation.
	mFV_DoOK: open it anyway.
	mFV_DoShowDialog: show dialog box and let user decide.
mFS_MakeIconic	Make document an icon as soon as it's opened (UNIX and Windows only).
	False: open file in an open window.
	True: iconify it.

Property	Instruction or situation and possible values
mFS_MakeVisible	Make document or book visible as soon as it's opened. Note that invisible documents and books can never be active. True: make visible. False: don't make visible.
mFS_NameStripe	String specifying the name that appears on the document title bar. Setting this property on Macintosh platforms has no effect. Empty string
mFS_NewDoc	Create new document. False: open existing document. True: create a new document.
mFS_OpenAsType	Specify the format of the file to import. mFVAUTORECOGNIZE: Default value; recognize the file type aoutmatically. mFVTYPE_BINARY: A FrameMaker binary file. mFVTYPE_MIF mFVTYPE_TEXT mFVTYPE_SGML mFVTYPE_XML mFVTYPE_FILTER: Use a filter to import this file. You must specify a valid file type hint for mFS_FileTypeHint.
mFS_OpenBookViewOnly	Open book in View Only format. False: don't open in View Only format. True: open in View Only format.
mFS_OpenDocViewOnly	Open document in View Only format. False: don't open in View Only format. True: open in View Only format.
mFS_OpenId	Set this to the ID of the document in the current window if you want to open the new ocument in that window, and mFS_OpenInNewWindow is set to False. 0

Property	Instruction or situation and possible values
mFS_OpenFileNotWritable	How to handle the case when opening a file the client cannot write to. mFV_DoCancel: Cancel the open operation mFV_DoOK: Open the file in read-only format mFV_DoShowDialog: Display an alert to notify that the file is not writable.
mFS_OpenInNewWindow	Open file in new window. True: open file in new window. False: open file in the window occupied by the document specified by the FS_OpenId property.
mFS_RefFileNotFound	Document imports another file that isn't available. mFV_DoCancel: cancel Open operation. mFV_AllowAllRefFilesUnFindable: open anyway and ignore the referenced file. mFV_DoShowDialog: show dialog box and let user decide.
mFS_SgmlOpenApplication	String specifying the name of the structure application to use when opening an XML or SGML file. This parameter takes precedence over any other structure application specification. If the specified application doesn't exist, the calling method will fail. Empty string (No Application Used)
mFS_ShowBrowser	Display Open dialog box. False: don't display it. True: display it.
mFS_UpdateBrowserDirectory	Update directory displayed in browser dialog box. False: don't update it. True: update it.
mFS_UpdateTextReferences	Update text insets. mFV_DoUserPreference: update text insets if the document property, FP_DontUpdateTextInsets, is False. mFV_DoYes: update text insets. mFV_DoNo: don't update text insets.

Property	Instruction or situation and possible values
mFS_UpdateXRefs	Update cross-references.
	mFV_DoUserPreference: update cross-references if the document property, FP_DontUpdateXRefs, is False.
	mFV_DoYes: update cross-references.
	mFV_DoNo: don't update cross-references.
mFS_UseAutoSaveFile	Use Autosave file if it is present.
	mFV_DoCancel: cancel the Open operation.
	mFV_DoYes: use it.
	mFV_DoNo: don't use it.
	mFV_DoShowDialog: show dialog box and let user decide.
mFS_UseRecoverFile	Use Recover file if it is present.
	mFV_DoCancel: cancel Open operation.
	mFV_DoYes: use it.
	mFV_DoNo: don't use it.
	mFV_DoShowDialog: show dialog box and let user decide.

Returns

A list of import properties. If it fails, the list is empty.

Examples

See the examples for the OpenBook and OpenDocument methods.

GetPropVals

Returns a list of properties and their values for the current object.

Synopsis

```
GetPropVals()
```

Details

This function is useful when you are creating a new object that is very similar to an existing object of the same type. You can get the property values from the existing object, then apply them to the new object. After that, you only have to change a small set of the properties to distinguish the new object from the existing one.

Returns

An array of properties with their values set. If it fails, this method sets FA_errno to one of the following:

Error code	Meaning
mFE_BadParameter	The function call specified an invalid parameter
mFE_BadOperation	The function call specified an illegal operation

Examples

The following example gets the properties from a specific paragraph format. It then creates a new paragraph format and sets the properties to it. Finally, the code changes the font size and capitalization for the new format.

```
Dim oDoc As FMDocument
Dim oNewFmt As FMParagraphFormat
Dim props As Variant

Set oDoc = mySession.ActiveDoc
props = mySession.GetNamedObject("Heading2", mFO_PgfFmt, oDoc).GetPropVals
Set oNewFmt = oDoc.NewNamedObject("HeadingNew", mFO_PgfFmt)
oNewFmt.SetPropVals props
oNewFmt.FontSize = oNewFmt.FontSize * 2
oNewFmt.Capitalization = mFV_CAPITAL_CASE_SMALL
```

GetSaveDefaultParams

Returns a default property list that you can use to with the Save method.

Synopsis

```
GetSaveDefaultParams()
```

Details

This method returns a list of properties that you can use to script the behavior of the Save method. By changing the settings of these properties, you can determine what type of file to open and what to do under different circumstances. The following tables list the parameters, showing the default value for each, followed by the other values you can set.

Properties for All Open Operations:

Property	Instruction or situation and possible values
mFS_AlertUserAboutFailure	Alert user if an unexpected condition occurs. during the save operation False: don't notify user when unexpected conditions occur. True: notify user when unexpected conditions occur.
mFS_AutoBackupOnSave	Specifies whether to create a backup file. mFV_SaveUserPrefAutoBackup: follow preference specified by the session's mFP_AutoBackup property. mFV_SaveYesAutoBackup: make a backup. mFV_SaveNoAutoBackup: don't make a backup.
mFS_DontNotifyAPIClients	Don't notify other FrameAC plugins and FDK clients of the save operation. False: notify them. True: don't notify them.
mFS_FileType	Specifies the type of file to save to. This file type must be one that FrameMaker saves natively. Note that HTML and XML are saved via filters, and so you must specify a filter hint string via mFS_SaveFileTypeHint. mFV_SaveFmtBinary: save in Frame binary format for this version of FrameMaker. mFV_SaveFmtBinary60: save in binary format for FrameMaker 6.0 mFV_SaveFmtInterchange: save as MIF. mFV_SaveFmtViewOnly: save in View Only format mFV_SaveFmtSgml: save in SGML format. mFV_SaveFmtText: save in Text Only format. mFV_SaveFmtXml: save in XML format. mFV_SaveFmtPdf: save as PostScript, and then invoke Acrobat Distiller to create a PDF version of the document. This is the same as choosing PDF from the Format popup menu in the Save As dialog box. mFV_SaveFmtFilter: filter on save, using mFS_SaveFileTypeHint to determine the filter.

Property	Instruction or situation and possible values
mFS_FileIsInUse	Another user or session is recorded in the file's lock file. mFV_DoCancel: cancel the save operation. mFV_DoShowDialog: show dialog box and let user decide. mFV_ResetLockAndContinue: Attempt to reset the file lock and save the document
mFS_LockCantBeReset	The user clicked Save Anyway in the File In Use dialog box, or the value of mFS_FileInUse is set to mFV_ResetLockAndContinue, but the lock file can't be reset. This is usually due to permissions in the lock file. mFV_DoCancel: Cancel the save operation mFV_DoShowDialog: Display the Cannot Lock File dialog box mFV_DoOK: Save the document anyway
mFS_ModDateChanged	The file has changed since the last time it was opened or saved in the current session. Somebody else has probably modified the file. mFV_DoCancel: Cancel the save operation mFV_DoShowDialog: Display the File Has Changed alert box mFV_DoOK:Save the document anyway
mFS_SaveFileNotWritable	The file permissions will not allow the file to be saved. FV_DoCancel: Cancel the save operation FV_DoShowDialog: Display the Cannot Lock FIle alert box
mFS_SaveFileTypeHint	If FS_FileType is FV_SaveFmtFilter, this string enables the FrameMaker product to call the correct filter. For example, use 0001ADBEHTML to save as HTML or 0001ADBEXML to save as XML. For information on the syntax of this string, see "Graphic and Text Inset Hint Strings" on page 407. Empty string.

Property	Instruction or situation and possible values
mFS_MakePageCount	Specifies how to round the page count. mFV_UseCurrentSetting: use default specified by the document property, mFP_PageRounding. mFV_DontChangePageCount: leave pages as is. mFV_MakePageCountEven: with odd number of pages, add a page to end of document. mFV_MakePageCountOdd: with even number of pages, add a page to end of document. mFV_DeleteEmptyPages: remove extra pages at end of document.
mFS_RetainNameStripe	Specifies whether to change the name in document title bar to the name the file is saved to. False: change name in title bar to the name the file is saved to. True: do not change name in title bar.
mFS_SaveAsModeName	Specifies where to get filename if mFS_SaveMode set to mFV_ModeSaveAs. mFV_SaveAsNameProvided: save under the filename specified in saveAsName parameter of the Save method. mFV_SaveAsUseFileName: save as name shown on document title bar. mFV_SaveAsNameAskUser: prompt user for name.
mFS_SaveMode	Specifies whether to use Save or Save As mode. mFV_ModeSaveAs: use Save As mode. mFV_ModeSave: use Save mode.
mFS_SaveTextExtraBlankLineAtEOP	Specifies whether to add an extra line at the end of each paragraph if the file is being saved as Text Only. False: don't add an extra line. True: add an extra line.

Property	Instruction or situation and possible values
mFS_SaveTextTblSetting	Specifies how to deal with tables if the file is being saved as Text Only. FV_SaveTblUserPref: use setting last specified in Save as Text dialog box. FV_SaveTblRowsAsPgfs: save each table cell as a paragraph row-by-row. FV_SaveTblColsAsPgfs: save each table cell as a paragraph column-by-column. FV_SaveSkipTbls: omit tables from a Text Only file. FV_SaveTextTblCellSeparator: the character to write as a cell separator in the text file. FV_SaveTextTblRowColumnSeparator: the character to write as a row or column separator in the text file.
mFS_SgmlSaveApplication	String specifying the name of the structure application to use when saving an XML or SGML file. This parameter takes precedence over any other structure application specification. If the specified application doesn't exist, the calling method will fail. Empty string (No Application Used)
mFS_ShowSaveTextDialog	Specifies whether to display dialog box if the file is being saved in Text Only format. False: don't display dialog box. True: display dialog box asking user whether to put paragraph returns at the end of each line.
mFS_UpdateFRVList	Specifies whether the file will be added to the list of files recently visited that appears in the File menu. This is set to False by default. False: don't add the file to the list. True: add the file to the list.

Returns

A list of import properties. If it fails, the list is empty.

Examples

The following code gets the default Save paramaters, and modifies the parameters to save a document as ViewOnly. It then uses the parameters to save a ViewOnly version of the current document.

```
Dim oDoc As FMDocument, oViewDoc As FMDocument
```

```
Dim saveParams As Variant, returnParams As Variant


Set oDoc = mySession.ActiveDoc
saveParams = mySession.GetSaveDefaultParams
For i = 0 To (UBound(saveParams) - 1)
     Select Case saveParams(i).propIdent.num
          Case Is = mFS_FileType
               saveParams(i).propVal.Value = mFV_SaveFmtViewOnly
     End Select
Next i
Set oViewDoc = oDoc.Save("C:\tmp\ViewOnlyDoc.fm", saveParams, returnParams)
```

GetText

Gets text from an object.

Synopsis

```
GetText(flags As FDK_TextItemTypes,
Flags2 As FDK_TextItemTypes2)
```

Arguments

flags. Flags that specify the type of text items to retrieve. To get specific types of text items, OR the constants that represent them (for example, use a bitwise OR to combine `mFTI_FlowBegin` and `mFTI_String`) into `flags`. To get all types of text items, specify `-1`. If you don't want to get any of the types represented by the possible entries for flags, specify `0`. For a complete list of the constants that represent text item types, see the tables below

Flags2 Flags for the extended set of text item types that specify the type of text items to retrieve. To get specific types of text items from the extended set, OR the constants that represent them (for example, use a bitwise OR to combine `mFTI2_RubiTextBegin` and `mFTI2_RubiTextEnd`) into `Flags2`

Details

GetText returns a list of text items (udtTextItem). Each text item contains either a string of text, the ID of an object that appears within the text (such as a table or an anchored frame), an indicator that the text properties have changed, or the ID of an object that contains text (such as a paragraph or a text column). For more information on how FrameAC represents text, see See "How FrameMaker Represents Text" on page 413.

You can use this method to get the text from the following objects:

- FMCell
- FMCrossReference
- FMElement
- FMFlow
- FMFootnote
- FMParagraph
- FMSubCol
- FMTextFrame
- FMTextInset
- FMTextInset_ApiClient
- FMTextInset_Flow
- FMTextInset_Text
- FMTextInset_TextTable
- FMVariable

The GetText method takes two arguments, which specify the type of text to get. Two arguments are required because the addition of support for Japanese text objects in FrameMaker extended the number of item types beyond what can be represented in a single integer. In general, if your document doesn't include Japanese text, then you can always specify a `0` for the Flags2 argument. The following two tables list the text item types, and their associated data.

Text item types for the first `flags` argument:

Text item type (dataType)	What the text item represents	Text item data
`mFTI_CharPropsChange`	A change in the text properties	Flags indicating which properties have changed (see the table below)
`mFTI_ElementBegin`	The beginning of a container element	ID of an FMElement
`mFTI_ElementEnd`	The end of a container element	ID of an FMElement
`mFTI_ElemPrefixBegin`	The beginning of an element's prefix	ID of an FMElement
`mFTI_ElemPrefixEnd`	The end of an element's prefix	ID of an FMElement

Text item type (dataType)	What the text item represents	Text item data
mFTI_ElemSuffixBegin	The beginning of an element's suffix	ID of an FMElement
mFTI_ElemSuffixEnd	The end of an element's suffix	ID of an FMElement
mFTI_FlowBegin	The beginning of a flow	ID of an FMFlow
mFTI_FlowEnd	The end of a flow	ID of an FMFlow
mFTI_FnAnchor	A footnote	ID of an FMFootnote
mFTI_FrameAnchor	An anchored frame	ID of an FMAnchoredFrame
mFTI_LineBegin	The beginning of a line	Nothing
mFTI_LineEnd	The end of a line and the line end type	If the line end is a normal line end, 0; if it is a forced line end, the mFTI_HardLineEnd flag is set; if it is a hyphen line end, the mFTI_HyphenLineEnd flag is set
mFTI_MarkerAnchor	A marker	ID of an FMMarker
mFTI_PageBegin	The beginning of a page	ID of an FMBodyPage, FMHiddenPage, FMMasterPage, FMReferencePage
mFTI_PageEnd	The end of a page	of an FMBodyPage, FMHiddenPage, FMMasterPage, FMReferencePage
mFTI_PgfBegin	The beginning of a paragraph	ID of an FMParagraph
mFTI_PgfEnd	The end of a paragraph	ID of an FMParagraph
mFTI_RubiCompositeBegin	The beginning of a rubi composite (and the beginning of oyamoji text).	ID of an FMRubi
mFTI_RubiCompositeEnd	The end of a rubi composite.	ID of an FMRubi
mFTI_RubiTextBegin	The beginning of rubi text (and the end of oyamoji text).	ID of an FMRubi
mFTI_RubiTextEnd	The end of rubi text.	ID of an FMRubi
mFTI_String	A string of characters with the same condition and character format	A character string
mFTI_SubColBegin	The beginning of a column	ID of an FMSubCol
mFTI_SubColEnd	The end of a column	ID of an FMSubCol

Text item type (dataType)	What the text item represents	Text item data
mFTI_TblAnchor	A table	ID of an FMTable
mFTI_TextFrameBegin	The beginning of a text frame	ID of an FO_TextFrame
mFTI_TextFrameEnd	The end of a text frame	ID of an FMTextFrame
mFTI_TextInsetBegin	The beginning of a text inset	ID of an FMTextInset_ApiClient, FMTextInset_Flow, FMTextInset_Text, or FMTextInset_TextTable
mFTI_TextInsetEnd	The end of a text inset	ID of an FMTextInset_ApiClient, FMTextInset_Flow, FMTextInset_Text, or FMTextInset_TextTable
mFTI_TextObjId	The object to which the offsets of all the text items are relative	ID of an FMParagraph, FMCell, FMTextLine, FMTextInset_ApiClient, FMTextInset_Flow, FMTextInset_Text, or FMTextInset_TextTable
mFTI_VarBegin	The beginning of a variable instance	ID of an FMVariable
mFTI_VarEnd	The end of a variable instance	ID of an FMVariable
mFTI_XRefBegin	The beginning of a cross-reference instance	ID of an FMCrossReference
mFTI_XRefEnd	The end of a cross-reference instance	ID of an FMCrossReference

Text item types for the extended set, to be specified in `Flags2`:

Text item type (dataType)	What the text item represents	Text item data
mFTI2_RubiTextBegin	The beginning of rubi text (and implicitly, the beginning of rubi text	ID of the FMRubi object for the rubi composite that contains the rubi text
mFTI2_RubiTextEnd	The end of rubi text	ID of the FMRubi object for the rubi composite that contains the rubi text
mFTI2_RubiCompositeBegin	The beginning of a rubi composite (and implicitly, the beginning of oyamoji text)	ID of an FMRubi object

Text item type (dataType)	What the text item represents	Text item data
mFTI2_RubiCompositeEnd	The end of a rubi composite	ID of an FMRubi object

The following table lists the bit flags thatyou can bitwise AND with the data of an mFTI_CharPropsChange text item. For example, to determine if the font family changed, bitwise AND the mFTF_FAMILY flag with the data field:

Flag	Meaning
mFTF_ALL	OR of all the flags listed above.
mFTF_ANGLE	The font angle has changed.
mFTF_CAPITALIZATION	The capitalization has changed.
mFTF_CHANGEBAR	The change bars have changed.
mFTF_CHARTAG	The Character Catalog format has changed.
mFTF_COLOR	The color has changed.
mFTF_CONDITIONTAG	The condition tag has changed.
mFTF_ENCODING	The text encoding has changed.
mFTF_FAMILY	The font family has changed.
mFTF_IIF	An internal flag having to do with asian text. input. If there is a non-zero value for this flag, a front end processor is controlling that text; you should not modify the associated text item.
mFTF_KERNX	The kern-x characteristic has changed.
mFTF_KERNY	The kern-y characteristic has changed.
mFTF_LANGUAGE	Character language has changed
mFTF_OUTLINE	The outline characteristic has changed.
mFTF_OVERLINE	The overline characteristic has changed.
mFTF_PAIRKERN	The pair kerning has changed.
mFTF_POSITION	The character position has changed.
mFTF_SHADOW	The shadow characteristic has changed.
mFTF_SIZE	The font size has changed.
mFTF_SPREAD	The font spread has changed.
mFTF_STRETCH	Font stretch value has changed
mFTF_STRIKETHROUGH	The strikethrough characteristic has changed.
mFTF_TSUME	Tsume setting has changed
mFTF_UNDERLINING	The underlining has changed.
mFTF_VARIATION	The font variation has changed.

Flag	Meaning
mFTF_WEIGHT	The font weight has changed.

If you call GetText for a structured element (FMElement object), the returned information depends on the type of element, as shown in the following table:

Element type	Information returned by GetText
mFV_FO_CONTAINER	All the text items from the beginning to the end of the element.
mFV_FO_SYS_VAR	All the text items from the beginning to the end of the variable.
mFV_FO_XREF	All the text items from the beginning to the end of the cross-reference.
mFV_FO_FOOTNOTE	All the text items from the beginning to the end of the footnote.
mFV_FO_TBL_TITLE	All the text items from the beginning to the end of the table title.
mFV_FO_TBL_CELL	All the text items from the beginning to the end of the cell.
mFV_FO_TBL_HEADING	Nothing. GetText fails.
mFV_FO_TBL_BODY	Nothing. GetText fails.
mFV_FO_TBL_FOOTING	Nothing. GetText fails.
mFV_FO_MARKER	Nothing. GetText fails.
mFV_FO_TBL	Nothing. GetText fails.
mFV_FO_GRAPHIC	Nothing. GetText fails.
mFV_FO_EQN	Nothing. GetText fails.
mFV_FO_TBL_ROW	Nothing. GetText fails.

Returns

A list of text items — on failure, the list has a length of 0. On failure, the method sets one of the following error codes to FA_errno:

Error code	Meaning
mFE_NotTextObject	Object specified for the text location is not an object that contains text
mFE_BadOperation	The function call specified an illegal operation

Examples

The following code loops through all the tables in the document's main flow. It then displays the the first paragraph from the title in each table. Notice that it requires two separate calls to GetText — one to get all the tables, and another to get the title text. The code skips the second call to GetText for any table that does not have a title.

```
Dim oDoc As FMDocument
Dim oFlow As FMFlow
```

```
Dim oTbl As FMTable
Dim oPgf As FMParagraph
Dim ti_table As udtTextItem, ti_text As udtTextItem
Dim list_tables As Variant, list_text As Variant
Dim i As Long, c As Integer
Dim titleStr As String

Set oDoc = mySession.ActiveDoc
Set oFlow = oDoc.MainFlowInDoc

list_tables = oFlow.GetText(mFTI_TblAnchor, 0)
For i = 0 To UBound(list_tables)
     titleStr = ""
     ti_table = list_tables(i)
     Set oTbl = mySession.GetObject(ti_table.Value, oDoc)
     Set oPgf = oTbl.FirstPgf
     If Not oPgf Is Nothing Then
          list_text = oPgf.GetText(mFTI_String, 0)
          For c = 0 To UBound(list_text)
               ti_text = list_text(c)
               titleStr = titleStr + ti_text.Value
          Next c
          mySession.Alert titleStr, mFF_ALERT_CONTINUE_NOTE
     End If
Next i
```

GetTextForRange

Gets the text for a specified text range. It provides the same parameters and functionality as the GetText method, except it allows you to specify a text range instead of an object.

Synopsis

```
GetTextForRange(textRange,
flags As Long,
Flags2 As Long)
```

Arguments

textRange	The text range containing the text you want to get

flags	Flags that specify the type of text items to retrieve. To get specific types of text items, OR the constants that represent them (for example, use a bitwise OR to combine `mFTI_FlowBegin` and `mFTI_String`) into `flags`. To get all types of text items, specify `-1`. If you don't want to get any of the types represented by the possible entries for flags, specify `0`. For a complete list of the constants that represent text item types, see the tables below
Flags2	Flags for the extended set of text item types that specify the type of text items to retrieve. To get specific types of text items from the extended set, OR the constants that represent them (for example, use a bitwise OR to combine `mFTI2_RubiTextBegin` and `mFTI2_RubiTextEnd`) into `Flags2`

Details

Method description goes here.

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`FE_BadParameter`	The function call specified an invalid parameter
`FE_BadOperation`	The function call specified an illegal operation

Examples

The following code displays the text in the current selection.

```
Dim myText As String
Dim i As Long, c As Integer
Dim list_text As Variant
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
list_text = oDoc.GetTextForRange(oDoc.TextSelection, mFTI_String, 0)
myText = ""
For i = 0 To UBound(list_text)
     ti_text = list_text(i)
     myText = myText + ti_text.Value
Next i
mySession.Alert myText, mFF_ALERT_CONTINUE_NOTE
```

GetTextProps

Gets a list of property values for the character to the right of the specified text location.

Synopsis

```
GetTextProps(Location)
```

Arguments

Location The udtTextLoc that identifies a character position in text.

Details

You can only get text properties for a single character at a time. This is because a range of characters may contain any number of different properties for each character in the range. Contrast this limitation to the fact that you can set text properties to a range of text via the SetTextProps method.

Returns

An array of properties with their values set. If it fails, this method sets FA_errno to one of the following:

Error code	Meaning
mFE_BadParameter	The function call specified an invalid parameter
mFE_BadOperation	The function call specified an illegal operation

Examples

The following code removes any type of underlining from the selected text. This code only operates if the selection is within a single paragraph. This example is not of practical value because there are easy ways to get this effect via the FrameMaker user interface. However, the code shows how to loop through a range of text and check the properties for each character in the range.

```
Dim oDoc As FMDocument
Dim props As Variant
Dim f_tLoc As udtTextLoc, f_tRange As udtTextRange
Dim i As Long, c As Long

Set oDoc = mySession.ActiveDoc
f_tRange = oDoc.TextSelection

'Only do this if the range is within one pgf.
If Not f_tRange.beg.objId = f_tRange.end.objId Then Exit Sub
```

```
f_tLoc = f_tRange.beg
For i = f_tRange.beg.offset To f_tRange.end.offset - 1
     f_tLoc.offset = i
     props = oDoc.GetTextProps(f_tLoc)
     For c = 0 To (UBound(props) - 1)
     If props(c).propIdent.num = mFP_Underlining Then
          If Not props(c).propVal.Value = mFV_CB_NO_UNDERLINE Then
               props(c).propVal.Value = mFV_CB_NO_UNDERLINE
               f_tRange.end.offset = i + 1
               oDoc.SetTextProps f_tRange, props
          End If
     End If
     Next c
Next i
```

GetUpdateDefaultParams

a default property list that you can use to call the UpdateBook method.

Synopsis

```
GetUpdateBookDefaultParams()
```

Returns

A property list with the properties shown in the following table. The first value listed for each property is thedefault value that propert.

Update parameters property	Instruction or situation and possible values
`mFS_AlertUserAboutFailure`	Alert user with warnings and messages if necessary. `False`: don't notify user when unexpected conditions occur. `True`: notify user when unexpected conditions occur.

Update parameters property	Instruction or situation and possible values
mFS_AllowInconsistentNumProps	Allow the FrameMaker product to update numbering, text insets, etc. of all the FrameMaker documents in the book, even if there are documents in the book with numbering properties that don't match the properties specified in the book. mFVS_DoOK: update numbering even if there are inconsistent properties in the book mFVS_DoCancel: cancel the update operation when it encounters a document with inconsistent numbering properties mFVS_DoShowDialog: show dialog box and let user decide
mFS_AllowNonFMFiles	Allow the FrameMaker product to update numbering, text insets, etc. of all the FrameMaker documents in the book, even if there are documents in the book that were not created by a FrameMaker product. mFVS_DoOK: update the book even if the book contains files not created by a FrameMaker product mFVS_DoCancel: cancel the update operation when it encounters a document not created by a FrameMaker product mFVS_DoShowDialog: show dialog box and let user decide
mFS_AllowViewOnlyFiles	Allow the FrameMaker product to update view-only documents in the book. mFVS_DoOK: update the view-only documents mFVS_DoCancel: cancel the entire update operation when it encounters a view-only document mFVS_DoShowDialog: show dialog box and let user decide
mFS_MakeVisible	Make newly generated files (lists and indexes) visible. True: make visible False: don't make visible
mFS_ShowBookErrorLog	Display the book error log for this update operation. False: don't display the error log; all warnings and errors go to the console True: display the error log

Update parameters property	Instruction or situation and possible values
mFS_UpdateBookGeneratedFiles	Update generated files such as TOC, lists, and indexes. Only update those generated files that have mFP_GenerateInclude set to True in their associated FMBookComponent objects. True: update generated files False: don't update generated files
mFS_UpdateBookNumbering	Update numbering in all the book's documents. True: update numbering False: don't update numbering

Examples

See the example for the UpdateBook method.

Import

Imports text or graphics into a document.

Synopsis

```
Import(TextLocation,
filename As String,
ImportParamters,
ReturnParameters) As FMObject
```

Arguments

TextLocation	The text location at which to import the file
filename	The full pathname of the file to import
ImportParameters	A property list telling FrameMaker how to import the file, and how to respond to errors and other conditions. To get the default set of properties, use the GetImportDefaultParams method.
ReturnParameters	A property list that provides information about how the FrameMaker product imported the file

Details

The Import method allows you to specify a script (property list) telling FrameMaker how to import text or graphics and how to deal with error and warning conditions. For example, you can specify whether to import a file by reference or by copy.

If you import a file by reference, the Import method creates an inset. The following table summarizes the types of files you can import with this method and the types of inset objects it creates when you import them by reference.

File type	Type of inset object Import creates
Graphics	FMInset
Text	FMTextInset_Text or FMTextInset_TextTable
FrameMaker binary document	FMTextInset_Flow
MIF	FMTextInset_Flow

When importing a graphic, you can specify that it be imported at its default resolution by setting the `mFS_GraphicDpi` property to `0` and setting the `mFS_FitGraphicInSelectedRect` property to `False`. If the graphic has no default resolution, it is imported at 72 dpi.

Returns

The inset object that the import method creates, or Nothing if the parameters specify that an import by copy. If the Import method fails it assigns one of the following error codes to FA_errno:

Error code	Meaning
`mFE_SystemError`	System error, such as an unreadable file or insufficient memory
`mFE_BadParameter`	The property list contained an invalid parameter
`mFE_BadFileType`	The specified file exists, but it does not have the correct file type
`mFE_MissingFile`	The specified file does not exist
`mFE_NoSuchFlow`	The script specifies an import flow that does not exist
`mFE_FailedState`	Internal error
`mFE_CircularReference`	Importing the specified file causes a circular reference
`mFE_FileClosedByClients`	The file was closed by a client before it could be imported

The `ReturnParameters` list has the following properties, which indicate the status of the operation:

Property	Meaning and Possible Values
`mFS_ImportedFileName`	A string specifying the source file's pathname. If you scripted FS_ShowBrowser, this pathname can be different from the one you specified in the Import script.
`mFS_ImportNativeError`	The error condition; normally the same value as `FA_errno`. If the file is imported successfully, it is set to `mFE_Success`. See the table below for a list of possible values.
`mFS_ImportStatus`	A bit field indicating what happened when the file was imported. See the table below for a list of possible flags.

Both the mFS_ImportNativeError property and the FA_errno session property indicate the result of a call to the Import method. The following table lists the possible status flags and the FA_errno and mFS_ImportNativeError values associated with them:

mFS_ImportNativeError and FA_errno values	Possible mFS_ImportStatus flags
mFE_BadParameter, mFE_BadFileType, mFE_MissingFile, mFE_FailedState, or mFE_CanceledByClient (file was not imported)	mFV_BadImportFileName: the specified source filename is invalid mFV_BadImportFileType: the Import script specified a file type different from the source file's actual type mFV_BadImportScriptValue: the Import script contained an invalid property value mFV_BadTextFileTypeHint: The file was a text file, and the string in FS_FileTypeHint was not a valid import hint string. For information on the syntax of this string, see "Graphic and Text Inset Hint Strings" on page 407. mFV_MissingScript: The Import method was called without a script. mFV_DisallowedImportType: source file's type disallowed by script. mFV_NoMainFlow: script specified to import the main flow, but the source file does not have a main flow. mFV_NoFlowWithSpecifiedName: script specified a flow name that does not exist mFV_InsertionPointNotInText: the insertion point in the enclosing document is not in text mFV_InsufficientMemory: there is insufficient memory to import the source file mFV_BadEnclosingDocId: there is no open document with the specified ID mFV_ImportFileNotReadable: the specified source file is unreadable
mFE_Success	mFV_ImportedByCopy: the source file was imported by copy mFV_ImportedText: the source file is a text file mFV_ImportTextTable: the source file is a text file, which was imported into a table. mFV_ImportedMIF: the source file is a MIF file mFV_ImportedMakerDoc: the source file is a FASL file. mFV_ImportedFilteredFile: the source file was filtered

mFS_ImportNativeError and FA_errno values	Possible mFS_ImportStatus flags
	mFV_ImportedGraphicFile: the source file is a graphics file
	mFV_ImportedSgmlDoc: the source file is an SGML document
	mFV_ImportedXmlDoc: the source file is an XML document
mFE_Cancel	mFV_CancelFileText: the file is text, so the user or the Import script canceled the Import operation
	mFV_CancelFileGraphic: the source file is a graphic, so the user or the Import script canceled the Import operation
	mFV_CancelFileMacEdition: the source file is a Macintosh Edition, so the Import script canceled the Import operation
	mFV_CancelFileDoc: the file is a FASL file, so the user or the script canceled the Import operation
	mFV_CancelFileSgml: the file is an SGML document, so the user or the script canceled the Import operation
	mFV_CancelFileXml: the file is an XML document, so the user or the script canceled the Import operation
	mFV_CancelFileMIF: the source file is a MIF file, so the user or the script canceled the Import operation
	mFV_CancelFileFilterable: the source file is a filterable file, so the user or the script canceled the Import operation
	mFV_InsertionPointInFootnote: the insertion point was in a footnote and the import script specified to import the file as a table, so the file could not be imported
	mFV_InsertionPointInTableCell: the insertion point was in a table cell and the import script specified to import the file as a table, so the file could not be imported
	mFV_UserCanceledImport: the user canceled the Import operation
	mFV_UserCanceledImportBrowser: the user canceled the Import browser

Examples

The following code imports a text file. It also sets the import parameters to insure you don't import a FrameMaker file.

```
Dim oDoc As FMDocument

Dim i As Long

Dim importParams As Variant, returnParams As Variant
```

```
Set oDoc = mySession.ActiveDoc
importParams = mySession.GetImportDefaultParams
'Change properties to disallow Frame documents.
For i = 0 To (UBound(importParams) - 1)
     Select Case importParams(i).propIdent.num
          Case Is = mFS_DisallowMIF
               importParams(i).propVal.Value = True
          Case Is = mFS_DisallowDoc
               importParams(i).propVal.Value = True
          Case Is = mFS_DisallowGraphicTypes
               importParams(i).propVal.Value = True
     End Select
Next i
oDoc.Import oDoc.TextSelection.beg, "C:\tmp\mydata.txt", importParams, returnParams
```

ImportElementDefs

Imports element definitions and the format change list catalog from an EDD or structured FrameMaker document or book to a structured FrameMaker document or book.

Synopsis

```
ImportElementDefs(SourceDocOrBook As FMObject,
flags As Long) As Long
```

Arguments

SourceDocOrBook The FMDocument or FMBook object from which to import the element definitions

flags. Flags that specify how to import element definitions.

Details

When you import element definitions, the following flags specify parameters that control the import element definitions operation:

Flag	Meaning
mFF_IED_REMOVE_OVERRIDES	Clear format overrides.
mFF_IED_REMOVE_BOOK_INFO	If SourceDocOrBook specifies a document, clear formatting inherited from a parent book.

Flag	Meaning
mFF_IED_DO_NOT_IMPORT_EDD	If the document specified by SourceDocOrBook is an EDD, don't treat it as an EDD; just import its element catalog.
mFF_IED_NO_NOTIFY	Do not issue the mFA_Note_PreImportElemDefs or mFA_Note_PostImportElemDefs notifications.

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_WrongProduct	Current product interface isn't Structured FrameMaker

Examples

The following code imports the element catalog from a source document into the active document. This code assumes the source document object already exists.

```
Dim oDoc As FMDocument, oSourceDoc As FMDocument


Set oDoc = mySession.ActiveDoc
oDoc.ImportElementDefs oSourceDoc, mFF_IED_REMOVE_BOOK_INFO
```

ImportFormats

Imports formats from a document to a document or a book. If you import formats to a book, this method imports formats to each book component for which the mFP_ImportFmtInclude property is set to True.

Synopsis

```
ImportFormats(SourceDocOrBook As FMObject,
flags As Long) As Long
```

Arguments

SourceDocOrBook	The FMDocument or FMBook object from which to import the formats
flags.	Flags that specify how to import formats

Details

Method description goes here.

When you import formats, the following flags specify parameters that control the import operation:

Flag	Meaning
mFF_UFF_COLOR	Import colors

Flag	Meaning
mFF_UFF_COMBINED_FONTS	Import combined fonts
mFF_UFF_COND	Import conditions
mFF_UFF_DOCUMENT_PROPS	Import document properties
mFF_UFF_FONT	Import Character Catalog formats
mFF_UFF_MATH	Import equation settings
mFF_UFF_PAGE	Import page layouts
mFF_UFF_PGF	Import Paragraph Catalog formats
mFF_UFF_REFPAGE	Import reference pages
mFF_UFF_REMOVE_EXCEPTIONS	Remove exception formats from target documents
mFF_UFF_REMOVE_PAGE_BREAKS	Remove all forced page breaks from target documents
mFF_UFF_TABLE	Import Table Catalog formats
mFF_UFF_VAR	Import variable formats
mFF_UFF_XREF	Import cross-reference formats

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_Canceled	User canceled the operation
mFE_FailedState	The FrameMaker product failed to open one or more of the book's document files during the import operation

Examples

The following code imports cross-reference formats from a source document into the active document. This code assumes the source document object already exists.

```
Dim oDoc As FMDocument, oSourceDoc As FMDocument


Set oDoc = mySession.ActiveDoc
oDoc.ImportFormats oSourceDoc, mFF_UFF_XREF
```

Initialise

Establishes a connection with FrameMaker and initialises a FrameAC session.

Synopsis

```
Initialise(AuthObject As Object) As Long
```

Arguments

AuthObject A FrameMaker authorsation object

Details

> **Important:** Do not confuse this Initialise method with the creator Sub routine that you must include in every FrameAC plug-in. For more information about creating FrameAC plug-ins, see Chapter 2, “FrameAC Tutorial.”

Returns

A value of `1` if it succeeds.

Examples

The following code establishes a connection with FrameMaker. If there is no connection, the code exits from the subroutine.

```
Dim mySession As FMSession
Dim myAuth As New FrameAuthorise
Dim myEx As New FrameEx

Set mySession = facEx.Session
If Not mySession.Initialise(facAuth) = 1 Then
     Exit Sub
End If
```

IsCellEmpty

Determines whether a table cell is empty.

Synopsis

```
IsCellEmpty() As Long
```

Returns

`True` if the cell is empty.

Examples

The following code loops through a table and writes a string to every empty cell in a body row.

```
Dim i As Long, c As Long
Dim rowCells As Variant
Dim f_tLoc As udtTextLoc
```

```
Dim oTbl As FMTable, oRow As FMRow
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
Set oTbl = oDoc.SelectedTbl
Set oRow = oTbl.GetFirstRow(mFV_ROW_BODY)
For i = 0 To (oTbl.GetCountOfRowsOfType(mFV_ROW_BODY) - 1)
     rowCells = oRow.Cells
     For c = 0 To (UBound(rowCells))
          If rowCells(c).IsCellEmpty Then
               f_tLoc.objId = rowCells(c).FirstPgf.Id
               f_tLoc.offset = 0
               oDoc.AddText f_tLoc, "Empty Cell"
          End If
     Next c
     Set oRow = oRow.NextRowInTbl
Next i
```

IsRowEmpty

Determines whether a table row is empty.

Synopsis

```
IsRowEmpty() As Long
```

Returns

`True` if the row is empty.

Examples

The following code deletes all the empty rows in the table.

```
Dim i As Long
Dim oTbl As FMTable, oRow As FMRow, oTmp As FMRow
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
Set oTbl = oDoc.SelectedTbl
Set oRow = oTbl.GetFirstRow(mFV_ROW_BODY)
For i = 0 To (oTbl.GetCountOfRowsOfType(mFV_ROW_BODY) - 1)
```

```
    Set oTmp = oRow
    Set oRow = oRow.NextRowInTbl
    If oTmp.IsRowEmpty Then oTmp.Delete
Next i
```

MakeTblSelection

Selects a range of cells in a table.

Synopsis

```
MakeTblSelection(topRow As Long,
bottomRow As Long,
leftCol As Long,
rightCol As Long) As Long
```

Arguments

topRow	The number of the top row to select in the table
bottomRow	The number of the bottom row to select in the table
leftCol	The number of the left column to select in the table
rightCol	The number of the right column to select in the table

Details

Rows are counted from zero. The count of rows begins at the top row, whether it is a heading row or not, and it ends with the last row of the table, whether it is a footing row or not. If you specify a row that is not in the table, the function does nothing. Also, you can only select rows of the same type — you cannot extend a selection from the heading row into the body rows. For example, assume a table that has two heading rows. If you specify `0` for `topRow` and `2` for `bottomRow`, the method fails because there are not three heading rows. In that case, the method does nothing, and the table selection does not change.

Columns are also counted from zero, from left to right. As with rows, if you specify a columns that are out of range, the method fails and makes no change to the current table selection.

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_BadParameter`	The function call specified an invalid parameter
`mFE_BadOperation`	The function call specified an illegal operation

Examples

The following code selects the cells that are in the first column of the current table.

```
Dim hRowCount As Long, selectCount As Long
Dim oTbl As FMTable
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
Set oTbl = oDoc.SelectedTbl
hRowCount = oTbl.GetCountOfRowsOfType(mFV_ROW_HEADING)
selectCount = oTbl.GetCountOfRowsOfType(mFV_ROW_BODY) + hRowCount - 1
oTbl.MakeTblSelection hRowCount, selectCount, 0, 0
```

MenuItemInMenu

Determines whether a menu item or menu is on the menu or menu bar represented by the current FMMenu.

Synopsis

```
MenuItemInMenu(Menu As FMMenu,
recursive As Long) As FMMenu
```

Arguments

Menu	The menu item to search for
recursive	Specifies whether to search the submenus of the current FMMenu object. Specify True to seach recursively.

Returns

If it finds the menu or command you are searching for, this method returns the FMMenu object on which it found the menu or command. If the menu or command is not found, the method returns Nothing. The returned FMMenu object may be the same as the object from which you call this method. That would result if the found command or menu is not within a submenu.

If this method fails, it returns Nothing and assigns one of the following values to `FA_errno`:

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_BadParameter`	The function call specified an invalid parameter
`mFE_BadOperation`	The function call specified an illegal operation

Examples

The following code determines whether the Paste command is on the Edit menu.

```
Dim oDoc As FMDocument
Dim oMenu As FMMenu, oOnMenu As FMMenu, oCmd As FMCommand

Set oMenu = mySession.GetNamedObject("EditMenu", mFO_Menu, oDoc)
Set oCmd = mySession.GetNamedObject("Paste", mFO_Command, Nothing)
Set oOnMenu = oCmd.MenuItemInMenu(oMenu, True)
If oOnMenu Is Nothing Then
    mySession.Alert oCmd.Name + " is NOT on " + oMenu.Name, mFF_ALERT_CONTINUE_NOTE
Else
     mySession.Alert oCmd.Name + " IS on " + oMenu.Name, mFF_ALERT_CONTINUE_NOTE
End If
```

MergeIntoFirst

Merges the selected structural elements into the first element in the selection. Note that you must have at least two structural elements selected in the document when you call this method.

Synopsis

```
MergeIntoLast() As Long
```

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_BadParameter`	The function call specified an invalid parameter
`mFE_BadOperation`	The function call specified an illegal operation
`mFE_WrongProduct`	Current product interface isn't Structured FrameMaker

Examples

The following code merges the selected structural elements in the active document.

```
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
oDoc.MergeIntoFirst
```

MergeIntoLast

Merges the selected structural elements into the last element in the selection. Note that you must have at least two structural elements selected in the document when you call this method.

Synopsis

```
MergeIntoFirst() As Long
```

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_BadParameter`	The function call specified an invalid parameter
`mFE_BadOperation`	The function call specified an illegal operation
`mFE_WrongProduct`	Current product interface isn't Structured FrameMaker

Examples

The following code merges the selected structural elements in the active document.

```
Dim oDoc As FMDocument


Set oDoc = mySession.ActiveDoc
oDoc.MergeIntoLast
```

NewAnchoredFormattedObject

Creates the following types of anchored objects:
FMCrossReference
FMTable
FMVariable

Synopsis

```
NewAnchoredFormattedObject(ObjectType As FDK_ObjectDefs,
format As String,
TextLocation) As FMObject
```

Arguments

ObjectType	A constant value specifying the type of object to create
format	The format name
TextLocation	Where to insert the new anchored object

Details

This method inserts the object at the specified location in text. It uses arbitrary default properties for the new object. After you have created the object, you can then change its properties.

If you call this method to create a table, it uses the default numbers of rows and columns from the specified Table Catalog format. To use the default Table Catalog format for a new table, set `format` to `Nothing`. To specify the number of rows and columns when you create a table, use the NewTable method.

Returns

The inserted object if it succeeds, or `Nothing` on failure. If the method fails, it sets one of the following error codes to `mFA_errno`:

Error code	Meaning
`mFE_NotTextObject`	Object specified for text location is not a paragraph (`mFO_Pgf`)
`mFE_OffsetNotFound`	Offset specified for the text location couldn't be found in the specified paragraph or text line
`mFE_BadNew`	Object can't be created; the format specified by `format` may not exist

Examples

The following code adds a Current Date (Long) variable at the insertion point (or the beginning of the text selection) of the active document.

```
Dim oDoc As FMDocument


Set oDoc = mySession.ActiveDoc

oDoc.NewAnchoredFormattedObject mFO_Var, "Current Date (Long)",
     oDoc.TextSelection.beg
```

NewAnchoredObject

Creates the following types of anchored objects:
FMAnchoredFrame
FMFootnote
FMMarker
FMTextInset_ApiClient
FMTable

Synopsis

```
NewAnchoredObject(ObjectType As FDK_ObjectDefs,
TextLocation) As FMObject
```

Arguments

ObjectType A constant value specifying the type of object to create

TextLocation Where to insert the new anchored object

Details

This method inserts the object at the specified location in text. It uses arbitrary default properties for the new object. After you have created the object, you can then change its properties.

Tables created by this method have a single column and a single body row. It is usually easier to create new tables via the NewTable method.

Returns

The inserted object if it succeeds, or `Nothing` on failure. If the method fails, it sets one of the following error codes to `mFA_errno`:

Error code	Meaning
`mFE_NotTextObject`	Object specified for text location is not a paragraph (`mFO_Pgf`)
`mFE_OffsetNotFound`	Offset specified for the text location couldn't be found in the specified paragraph or text line
`mFE_BadOperation`	The function call specified an illegal operation

Examples

The following code inserts an anchored frame at the beginning of the current text selection of the active document.

```
Dim oDoc As FMDocument


Set oDoc = mySession.ActiveDoc
oDoc.NewAnchoredObject mFO_AnchoredFrame, oDoc.TextSelection.beg
```

NewBodyPage

Method Short description.

Synopsis

```
NewBodyPage(PreviousPage As FMBodyPage) As FMBodyPage
```

Arguments

PreviousPage The body page that your new page will come after. To insert a body page as the first page in the document, specify Nothing.

Details

When this method inserts a body page, the page is unconnected. That means that the page is not a part of the existing flow in eht document.

Returns

The new body page if it succeeds, or `Nothing` on failure. If the method fails, it sets the following error codes to `FA_errno`:

Error code	Meaning
`mFE_BadParameter`	The function call specified an invalid parameter
`mFE_BadOperation`	The function call specified an illegal operation

Examples

The following code creates a new unconnected body page as page 2 in the active document.

```
Dim oDoc As FMDocument


Set oDoc = mySession.ActiveDoc
oDoc.NewBodyPage oDoc.FirstBodyPageInDoc
```

NewBook

Creates a new book with the specified name.

Synopsis

```
NewBook(BookName As String) As FMBook
```

Arguments

BookName The name for the new book.

Details

The method creates a new book and uses `BookName` as the name to show in the book window's title bar. The new book has no book components in it.

This method does not save the book file to disk. If BookName specifies a valid path for the book file, the user can save the book to that location via Save or Save As. If BookName does not specify a valid path, Save and Save As operations open the Save As dialog box to the last saved location.

Returns

The new body page if it succeeds, or `Nothing` on failure. If the method fails, it sets the mfollowing error codes to `FA_errno`:

Error code	Meaning
`mFE_BadParameter`	The function call specified an invalid parameter
`FE_BadOperation`	The function call specified an illegal operation

Examples

The following code creates a new book file.

```
Dim o_newBook As FMBook


Set o_newBook = mySession.NewBook("C:\tmp\MyNewBook.book")
```

NewBookComponent

Inserts a book component in a FrameMaker book.

Synopsis

```
NewBookComponent(PreviousComponent As FMBookComponent) As
FMBookComponent
```

Arguments

`PreviousComponent` The FMBookComponent object that is previous to the new component. To make a first component in a book, use `Nothing`.

Details

When you add a new component to a book, there is no file associated with it. You must open a file, or create a new one and save it. Then you assign the document file's name to the book component's name property to associate the file with that book component.

Returns

A FMBookComponent object on success, or Nothing on failure. If the method fails, it assigns one of the following codes to FA_errno:

Error code	Meaning
`mFE_BadParameter`	The function call specified an invalid parameter
`mFE_BadOperation`	The function call specified an illegal operation

Examples

The following code creates a book file, then adds two document files to the book as book components.

```
Dim filename1 As String, filename2 As String
Dim templateName As String, bookName As String
Dim saveParams As Variant, returnParams As Variant
Dim oDoc1 As FMDocument, oDoc2 As FMDocument
Dim o_newBook As FMBook, o_bookComp As FMBookComponent

bookName = "C:\tmp\MyNewBook.book"
templateName = "C:\tmp\Test1.fm"
filename1 = "C:\tmp\MyFile1.fm"
filename2 = "C:\tmp\MyFile2.fm"
saveParams = mySession.GetSaveDefaultParams

Set oDoc1 = mySession.NewDocument(templateName, False)
oDoc1.Save filename1, saveParams, returnParams
Set oDoc2 = mySession.NewDocument(templateName, False)
oDoc2.Save filename2, saveParams, returnParams

Set o_newBook = mySession.NewBook(bookName)
Set o_bookComp = o_newBook.NewBookComponent(Nothing)
o_bookComp.Name = filename1
Set o_bookComp = o_newBook.NewBookComponent(o_bookComp)
o_bookComp.Name = filename2
```

NewBookComponentInHierarchy

Inserts a book component at a specified position in a FrameMaker structured book.

Synopsis

```
NewBookComponentInHierarchy(ComponentName As String,
ElementLocation) As FMBookComponent
```

Arguments

ComponentName The name of the file to be added to the book

ElementLocation udtElementLoc data to express where to add the new book component

Details

The book you are operating on must be structured.

When you add a new component to a book, you must open a file, or create a new one and save it. Then you pass the document file's name to this method to associate the file with the book component.

When the method adds a new component to the book, it doesn't specify the element definition for the new component. To specify the element type, get the book component's element via it's ComponentElement property, then modify the FMElement object as necessary.

Returns

A FMBookComponent object on success, or Nothing on failure. If the method fails, it assigns one of the following codes to FA_errno:

Error code	Meaning
mFE_BadCompPath	Component name specified for compName is invalid
mFE_BadNew	The object can't be created
mFE_BookUnStructured	Specified book is unstructured

Examples

The following code adds a new book component as the last component in the current structured book.

```
Dim oDoc1 As FMDocument
Dim o_book As FMBook
Dim f_elemLoc As udtElementLoc, f_elemRange As udtElementRange

Set o_book = mySession.ActiveBook

templateName = "C:\tmp\Test1.fm"
filename1 = C:\tmp\MyFile1.fm"
saveParams = mySession.GetSaveDefaultParams

Set oDoc1 = mySession.NewDocument(templateName, False)
oDoc1.Save filename1, saveParams, returnParams
f_elemLoc.parentId = o_book.HighestLevelElement.Id
f_elemLoc.childId = 0
f_elemLoc.offset = 0
Set o_bookComp = o_book.NewBookComponentInHierarchy(oDoc1.Name, f_elemLoc)
```

NewChild

Creates a new graphic object that is the child of (contained in) a frame object.

Synopsis

```
NewChild(objType As FDK_ObjectDefs) As FMGraphic
```

Arguments

ObjType The type of graphic object to create — for example, `mFO_TextFrame`

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mE_BadParameter`	The function call specified an invalid parameter
`mFE_BadOperation`	The function call specified an illegal operation

Examples

The following code loops through each page in the document. For each page, it finds the page frame, and creates a new text frame as a child of the page frame. It then adds text to the new text frame. Note that in practical code, you would have to size and place the new text frame.

```
Dim f_tLoc As udtTextLoc
Dim oTFrame As FMTextFrame
Dim o_BodyPage As FMBodyPage, o_PageFrame As FMPageFrame
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
Set o_BodyPage = oDoc.FirstBodyPageInDoc
Set o_PageFrame = o_BodyPage.PageFrame
Do Until o_PageFrame Is Nothing
     ' Loop pages, add the Text Frame, and some text.
     Set oTFrame = o_PageFrame.NewChild(mFO_TextFrame)
     oTFrame.Height = oTFrame.Height * 3
     oTFrame.Width = oTFrame.Width * 3

     f_tLoc.objId = oTFrame.FirstPgf.Id
     f_tLoc.offset = 0
     oDoc.AddText f_tLoc, "Repeated Text!"
```

```
     Set o_BodyPage = o_BodyPage.PageNext
     If o_BodyPage Is Nothing Then
          Set o_PageFrame = Nothing
     Else
          Set o_PageFrame = o_BodyPage.PageFrame
     End If
Loop
```

NewDocument

Creates a new document, using the specified file as a template.

Synopsis

```
NewDocument(TemplateFilename As String,
interactive As Long) As FMDocument
```

Arguments

TemplateFilename The full path to a document to use as a template

interactive Determines whether to open the New Document dialog box when creating the new file. `True` opens the dialog box.

Details

The method creates a new document and uses `TemplateFileName` to identify the template for the new document. The new document is untitled — you must use the Save method to save this document.

Returns

The new body page if it succeeds, or `Nothing` on failure. If the method fails, it sets the mfollowing error codes to `FA_errno`:

Error code	Meaning
`mFE_BadParameter`	The function call specified an invalid parameter
`FE_BadOperation`	The function call specified an illegal operation

Examples

The following code creates a new document and saves it.

```
Dim oDoc As FMDocument

Set oDoc = mySession.NewDocument("C:\tmp\Template1.fm", False)
```

```
oDoc1.Save "C:\tmp\MyFile1.fm", saveParams, returnParams
```

NewElement

Creates an FMElement object in a structured FrameMaker document at the specified text location, using the specified element definition.

Synopsis

```
NewElement(ElementDefinition As FMElementDefinition,
TextLocation) As FMElement
```

Arguments

ElementDefinition The FMElementDefinition object for the new element's definition

TextLocation The udtTextLoc to specify where to add the new element

Details

For object (noncontainer) elements, this method inserts the appropriate type of document object for the element. If there is a matching format rule, the method uses it to format the object. Otherwise, it uses one of the following default formats:

Object type	Object inserted	Format used if no matching format rule exists
mFV_FO_XREF	Cross-reference	Undefined XRef
mFV_FO_EQN	Equation	medium
mFV_FO_MARKER	Marker	Type 11
mFV_FO_TBL	Table with the format and number of rows and columns specified by the table format	Format A if it exists; otherwise, a table with a heading row, 8 body rows, a footing row, and 5 columns
mFV_FO_SYS_VAR	Variable	Filename (Long)
mFV_FO_GRAPHIC	A centered 1.0-inch by 1.0-inch anchored frame below the current position; cropped is off, and floating is on	

Returns

The new FMElement object, or Nothing on failure. On failure, the method sets one of the following error codes to FA_errno:

Error code	Meaning
mFE_BadNew	Object can't be created
mFE_NotTextObject	Object specified for the text location isn't a paragraph (FO_Pgf) or a text line (FO_TextLine)

Error code	Meaning
mFE_OffsetNotFound	Offset specified for the text location couldn't be found in the specified paragraph or text line

Examples

The following code inserts a new Head element at the beginning of the current text selection.

```
Dim oElem As FMElement
Dim oElemDef As FMElementDefinition
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
Set oElemDef = mySession.GetNamedObject("Head", mFO_ElementDef, oDoc)
Set oElem = oDoc.NewElement(oElemDef, oDoc.TextSelection.beg)
```

NewElementInHierarchy

Creates a structural element (FO_Element object) at a specified location in the element hierarchy of a structured FrameMaker document or book.

Synopsis

```
NewElementInHierarchy(ElementDefinition As FMElementDefinition,
ElementLocation) As FMElement
```

Arguments

ElementDefinition The FMElementDefinition object for the new element's definition

ElementLocation The udtElementLoc to specify where in the document hierarchy to add the new element

Details

To create the root element for a book, you must use NewElementInHierarchy — you can't use NewElement. Once a book is structured, you can add new book component elements via NewBookComponentInHierarchy.

For documents, you can't use F_ApiNewElementInHierarchy to add elements to an unstructured document. You must structure the document first by adding a root element with NewElement.

For object (noncontainer) elements, this method inserts the appropriate type of document object for the element. If there is a matching format rule, the method uses it to format the object. Otherwise, it uses one of the following default formats:

Object type	Object inserted	Format used if no matching format rule exists
mFV_FO_XREF	Cross-reference	Undefined XRef
mFV_FO_EQN	Equation	medium
mFV_FO_MARKER	Marker	Type 11
mFV_FO_TBL	Table with the format and number of rows and columns specified by the table format	Format A if it exists; otherwise, a table with a heading row, 8 body rows, a footing row, and 5 columns
mFV_FO_SYS_VAR	Variable	Filename (Long)
mFV_FO_GRAPHIC	A centered 1.0-inch by 1.0-inch anchored frame below the current position; cropped is off, and floating is on	

Returns

The new FMElement object, or `Nothing` on failure. On failure, the method sets one of the following error codes to `FA_errno`:

Error code	Meaning
mFE_BadNew	Object can't be created
mFE_NotTextObject	Object specified for the text location isn't a paragraph (FO_Pgf) or a text line (FO_TextLine)
mFE_OffsetNotFound	Offset specified for the text location couldn't be found in the specified paragraph or text line

Examples

The following code inserts a new Head element at the beginning of the current element selection.

```
Dim oElemDef As FMElementDefinition
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
Set oElemDef = mySession.GetNamedObject("Head", mFO_ElementDef, oDoc)
Set oElem = oDoc.NewElementInHierarchy(oElemDef, oDoc.ElementSelection.beg)
```

NewFormatChangeList

Creates a new format change list to add to a format rule clause.

Synopsis

```
Method Synopsis.
```

Details

> **Important:** To create a named format change list in the format change list catalog, use the NewNamedObject method.

Returns

The new FMFormatRuleClause, or Nothing on failure. On failure the method sets one of the following codes to `FA_errno`:

Error code	Meaning
`mFE_BadNew`	Object can't be created
`mFE_BadPropNum`	The property number specified for property is invalid
`mFE_WrongProduct`	Current product interface isn't Structured FrameMaker

Examples

For an example, see the code example for NewFormatRuleClause.

NewFormatRule

Adds a new format rule to the list of format rules in the current FMElementDefinition or FMFormatRuleClause object. If format rules already exist in the element definition or format rule clause, the new format rule is added at the end of the list.

Synopsis

```
NewFormatRule() As FMFormatRule
```

Details

You create format rules as members of FMElementDefinition or FMFormatRuleClause objects. A format rule can contain FMFormatRuleClause objects.

Returns

The new FMFormatRuleClause, or Nothing on failure. On failure the method sets one of the following codes to `FA_errno`:

Error code	Meaning
`mFE_BadNew`	Object can't be created
`mFE_BadPropNum`	The property number specified for property is invalid

Error code	Meaning
mFE_WrongProduct	Current product interface isn't Structured FrameMaker

Examples

For an example, see the code example for NewFormatRuleClause.

NewFormatRuleClause

Adds a new format rule clause to the list of clauses in the current format rule. If clauses already exist in the format rule, the new clause is added at the end of the list.

Synopsis

```
NewFormatRuleClause() As FMFormatRuleClause
```

Details

You create format rule clauses as members of FMFormatRule objects. A format rule clause can contain FMFormatRule objects.

Returns

The new FMFormatRuleClause, or Nothing on failure. On failure the method sets one of the following codes to FA_errno:

Error code	Meaning
mFE_BadNew	Object can't be created
mFE_BadPropNum	The property number specified for property is invalid
mFE_WrongProduct	Current product interface isn't Structured FrameMaker

Examples

The following code creates a new element definition, and adds a format rule and two format rule clauses to that element definition. One of the format rule clauses includes a format change list, which specifies SmallCap capitalization.

```
Dim oElem As FMElement
Dim oFmtRule As FMFormatRule, oFmtRuleClause As FMFormatRuleClause
Dim oFmtChangeList As FMFormatChangeList
Dim oElemDef As FMElementDefinition
Dim elemName As String
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
elemName = "MyNewElemDef"
```

```
' Create the new element definition, or get it if already exists.
Set oElemDef = mySession.GetNamedObject(elemName, mFO_ElementDef, oDoc)
If oElemDef Is Nothing Then _
     Set oElemDef = oDoc.NewNamedObject(elemName, mFO_ElementDef)
If oElemDef Is Nothing Then Exit Sub

' Specify the element definition
oElemDef.ElementDefType = mFV_FO_CONTAINER
oElemDef.ValidHighestLevel = False
oElemDef.ElementInCatalog = True
oElemDef.GeneralRule = "<TEXT>"

' Set up the format rule
Set oFmtRule = oElemDef.NewFormatRule(mFP_PrefixRules)
oFmtRule.FmtRuleType = mFV_CONTEXT_RULE
Set oFmtRuleClause = oFmtRule.NewFormatRuleClause
oFmtRuleClause.IsTextRange = True
Set oFmtRuleClause = oFmtRule.NewFormatRuleClause
oFmtRuleClause.Specification = "In all contexts"
oFmtRuleClause.ElemPrefixSuffix = "MyPrefixString: "
Set oFmtChangeList = oFmtRuleClause.NewFormatChangeList
oFmtChangeList.Capitalization = mFV_CAPITAL_CASE_SMALL
```

NewNamedObject

Creates a new object that is identified by a name.

Synopsis

```
NewNamedObject(ObjectName As String,
ObjectType As FDK_ObjectDefs) As FMObject
```

Arguments

ObjectName	The name of the object
ObjectType	A constant representing the type of object to create

Details

This method can create an object of the following types:

- FMBook — mFO_Book
- FMCharacterFormat — mFO_CharFmt
- FMColor — mFO_Color
- FMCombinedFontDefinition — mFO_CombinedFontDefn
- FMCommand — Command
- FMConditionalFormat — mFO_CondFmt
- FMCrossReferenceFormat — mFO_XRefFmt
- FMElementDefinition — mFO_ElementDef
- FMFormatChangeList — mFO_FmtChangeList
- FMMasterPage — mFO_MasterPage
- FMMenu — mFO_Menu
- FMMenuItemSeperator — mFO_MenuItemSeparator
- FMParagraphFormat — mFO_PgfFmt
- FMReferencePage — mFO_RefPage
- FMRulingFormat — mFO_RulingFmt
- FMTableFormat — mFO_TblFmt
- FMVariableFormat — mFO_VarFmt

This method uses arbitrary default properties for the objects it creates. After you have created the object, you can then change the properties.

> **Important:** When you create a new element definition, it does not appear in the Element Catalog unless you set the mFP_ElementInCatalog property to True.
>
> When you create a new book and specify a pathname, you must specify an absolute pathname for the ObjectName argument. To create an untitled book, pass an empty string for ObjectName.

Returns

The inserted object if it succeeds, or Nothing on failure. If the method fails, it sets one of the following error codes to mFA_errno:

Error code	Meaning
mFE_BadName	Specified name for the new object is invalid
mFE_BadNew	Object can't be created
mFE_DupName	Specified name for the new object belongs to an existing object

Examples

The following code creates a new paragraph format and a new master page.

```
Dim oNewPgfFmt As FMParagraphFormat
Dim oNewMasterPage As FMMasterPage
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
Set oNewPgfFmt = oDoc.NewNamedObject("MyNewPgfFmt", mFO_PgfFmt)
Set oNewMasterPage = oDoc.NewNamedObject("MyNewMasterPage", mFO_MasterPage)
```

NewParagraph

Creates a new paragraph in the document.

Synopsis

```
NewParagraph(PreviousItem As FMObject) As FMParagraph
```

Arguments

PreviousItem. The paragraph that is to precede the new paragraph. To add the paragraph at the start of a flow, specify the FMFlow object.

Returns

The inserted object if it succeeds, or `Nothing` on failure. If the method fails, it sets one of the following error codes to `mFA_errno`:

Error code	Meaning
`mFE_BadNew`	Object can't be created
`mFE_BadObjId`	Invalid object for `PreviousItem`

Examples

The following code creates two new paragraphs — one after the paragraph at the start of the current text selection, and another at the beginning of the current flow.

```
Dim oNewPgf As FMParagraph
Dim oCurrentFlow As FMFlow
Dim f_tRange As udtTextRange
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
f_tRange = oDoc.TextSelection
Set oNewPgf = oDoc.NewParagraph(mySession.GetObject(f_tRange.beg.objId, oDoc))
```

```
f_tRange.beg.objId = oNewPgf.Id
f_tRange.beg.offset = 0
oDoc.AddText f_tRange.beg, "Text in my new paragraph!"

Set oCurrentFlow = oNewPgf.InTextFrame.Flow
Set oNewPgf = oDoc.NewParagraph(oCurrentFlow)
f_tRange.beg.objId = oNewPgf.Id
oDoc.AddText f_tRange.beg, "Text at the start of the flow!"
```

NewTable

Creates a new FMTable object.

Synopsis

```
NewTable(format As String,
Cols As Long,
BodyRows As Long,
HeaderRows As Long,
FooterRows As Long,
TextLocation) As FMTable
```

Arguments

format	The table format tag. To use the default table format, pass an empty string.
Cols	The number of columns for the table. To use the default number from the table format catalog, specify `-1`.
BodyRows	The number of body rows for the table. To use the default number from the table format catalog, specify `-1`.
HeaderRows	The number of header rows for the table. To use the default number from the table format catalog, specify `-1`.
FooterRows	The number of footer rows for the table. To use the default number from the table format catalog, specify `-1`.
TextLocation.	The location at which to insert the new table's anchor. This location cannot be within a footnote, text line, or a table cell.

Details

When you create a table in the user interface, you can specify a Table Catalog format for the table. The FrameMaker product uses the following properties of the Table Catalog format as the defaults for the new table:

- Number of body rows (mFP_TblInitNumBodyRows)
- Number of columns (mFP_TblInitNumCols)
- Number of footer rows (mFP_TblInitNumFRows)
- Number of header rows (mFP_TblInitNumHRows)
- Paragraph formats for header, body, and footer cell

For example, if the Table Catalog format's mFP_TblInitNumCols property is set to 8, the FP_NumCols property of the new table is set to 8.

With this method, you can use the Table Catalog format properties as defaults for the number of rows and columns in a new table, or you can provide your own values.

After you have created a table, you can add or remove rows with AddRows and DeleteRows. You can add or remove columns with AddCols and DeleteCols. You can also change the table's other properties.

If you use this method to create a table in a structured FrameMaker document, FrameMaker applies default element tags, such as Table, Row, and Cell, to the table element and its child elements. To make these elements valid, you must add code to change their tags. In most cases it is easier to add tables to structured documents by calling NewElementInHierarchy or NewElement to add a table element.

Returns

The inserted object if it succeeds, or Nothing on failure. If the method fails, it sets one of the following error codes to mFA_errno:

Error code	Meaning
mFE_BadNew	Object can't be created
mFE_BadObjId	Invalid object for PreviousItem
mFE_NotTextObject	Object specified for the text location isn't a paragraph (FO_Pgf).
mFE_OffsetNotFound	Offset specified for the text location couldn't be found in the specified paragraph or text line.
mFE_BadOperation	Function call specified an illegal operation.

Examples

The following code creates a new table at the beginning of the current selection. The new table uses Format B, and has the number of rows and columns that are specified by the table format.

```
Dim oNewTbl As FMTable
Dim f_tRange As udtTextRange
Dim oDoc As FMDocument
```

```
Set oDoc = mySession.ActiveDoc
f_tRange = oDoc.TextSelection
Set oNewTbl = oDoc.NewTable("Format B", -1, -1, -1, -1, f_tRange.beg)
```

ObjectValid

Tests whether the object has been created correctly and is valid.

Synopsis

```
ObjectValid() As Long
```

Arguments

Arg Name. Arg Val

Returns

`True` if the object is valid, or false if it is not.

Examples

The following code creates an object, and tests whether the object is valid. If the object is not valid, the code exits the subroutine.

```
Dim oNewTbl As FMTable
Dim f_tRange As udtTextRange
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
f_tRange = oDoc.TextSelection
Set oNewTbl = oDoc.NewTable("Format B", -1, -1, -1, -1, f_tRange.beg)
If Not oNewTbl.ObjectValid Then
     Exit Sub
End If
```

OpenBook

Opens a book or creates a new book.

Synopsis

```
OpenBook(filename As String,
OpenParams,
OpenReturnParams) As FMBook
```

Arguments

filename	The full pathname of the book file to open.
OpenParams	A property list telling FrameAC how to open the book file and how to respond to errors and other conditions. To use the default list, specify `Nothing`.
OpenReturnParams	A property list that returns the filename and provides information

Details

To get a property list for `OpenParams`, use the GetOpenDefaultParams method. For a list of all the properties and values, see the documentation for that method. To create a new untitled book, set the `mFS_NewDoc` property to `True` before you use the OpenBook method.

Returns

The FMBook object that the OpenBook method creates, or `Nothing` on failure.

The `OpenReturnParams` list has the following properties, which indicate the status of the operation:

Property	Meaning and Possible Values
`mFS_OpenedFileName`	A string that specifies the opened file's pathname. If you scripted `mFS_ShowBrowser`, or the file was filtered, or you didn't specify the pathname, this pathname can be different from the one you specified in the Open script.
`mFS_OpenNativeError`	The error condition; normally the same value as `FA_errno`. If the file is opened successfully, it is set to `mFE_Success`. See the table below for a list of possible values.
`mFS_OpenStatus`	A bit field indicating what happened when the file was opened. See the table below for a list of possible flags.

Both the mFS_OpenNativeError property and the FA_errno session property indicate the result of a call to the Open method. The following table lists the possible status flags and the FA_errno and mFS_OpenNativeError values associated with them:

mFS_OpenNativeError and FA_errno values	Possible mFS_OpenStatus flags
mFE_BadParameter (file wasn't opened)	mFV_FileHadStructure: file had FrameMaker features, but current FrameMaker product is not FrameMaker
	mFV_FileAlreadyOpenThisSession: file is already open and script disallowed opening another copy
	mFV_BadFileType: file was an executable file or other unreadable type
	mFV_BadFileName: specified filename was invalid
	mFV_CantNewBooks: script specified a book that didn't exist (the Open operation can't create a new book)
	mFV_BadScriptValue: Open script contained an invalid property value
	mFV_MissingScript: OpenBook called without a script
	mFV_CantForceOpenAsText: Open script attempted to open the file as text, but file was wrong type
	mFV_DisallowedType: file was a Frame binary document and the Open script disallowed it
	mFV_DocDamagedByTextFilter: file was a text document and was damaged when it was filtered
	mFV_DocHeadersDamaged: the document headers were damaged (probably because of a file system problem)
	mFV_DocWrongSize: file is the wrong size (probably because of a file system problem)
	mFV_ChecksumDamage: bad checksum
mFE_SystemError (file wasn't opened)	mFV_TooManyWindows: too many windows were open
	mFV_BadTemplate: a bad template was specified
	mFV_FileNotReadable: don't have read permission for the file
mFE_Success (file was opened)	mFV_FileHasNewName: filename was changed from the name specified in the F_ApiOpen() call
	mFV_RecoverFileUsed: recover file was present, and it was used
	mFV_AutoSaveFileUsed: Autosave file was present, and the user or the Open script chose to use it
	mFV_FileWasFiltered: file was filterable and it was filtered

mFS_OpenNativeError and FA_errno values	Possible mFS_OpenStatus flags
	mFV_FontsWereMapped: the document contained unavailable fonts, which were mapped to substitute fonts
	mFV_FontMetricsChanged: the file contained fonts with changed metrics, but it was opened anyway
	mFV_FontsMappedInCatalog: the Paragraph or Character Catalog used unavailable fonts, which were mapped to substitute fonts
	mFV_LanguagesWerentFound: the document used some unavailable languages, but it was opened anyway
	mFV_BeefyDoc: the document file was extremely large (Macintosh platforms only), but it was opened anyway
	mFV_FileIsOldVersion: the file was from an old FrameMaker product version, but the user or the Open script chose to open it anyway
	mFV_FileStructureStripped: the file had FrameMaker features, which the user or the Open script chose to strip
	mFV_FileIsText: the file was a Text Only file, but the user or the Open script chose to open it anyway
	mFV_OpenedViewOnly: the user or the Open script chose to open the file as a View Only file
	mFV_EditableCopyOpened: the file was in use and the user or the Open script opened an editable copy
	mFV_BadFileRefsWereMapped: file reference contained illegal characters; the illegal characters were converted to something safe for the current platform
	mFV_ReferencedFilesWerentFound: imported graphics files couldn't be found, but the file was opened anyway
	mFV_FileAlreadyOpen: the file was in use and the user or the Open script opened another copy (Macintosh platforms only)
	mFV_UnresolvedXRefs: there were unresolved cross-references, but the file was opened anyway
	mFV_UnresolvedTextInsets: there were unresolved text insets, but the file was opened anyway

mFS_OpenNativeError and FA_errno values	Possible mFS_OpenStatus flags
mFE_Success, mFE_Canceled, mFE_FailedState, or mFE_CanceledByClient	mFV_LockWasReset: file lock was reset mFV_LockNotReset: file had lock that wasn't reset mFV_LockCouldntBeReset: file had lock that couldn't be reset mFV_FileWasInUse: file was in use mFV_FileIsViewOnly: file is a View Only file mFV_LockWasInvalid: file had invalid lock mFV_FileIsNotWritable: The file was not writable, and the user canceled the open via the alert. mFV_FileModDateChanged: The file has changed since the last time it was opened or saved in the current session.
mFE_Canceled (file wasn't opened)	mFV_CancelUseRecoverFile: a recover file was present, so the user or the Open script canceled the Open operation mFV_CancelUseAutoSaveFile: an Autosave file was present, so the user or the Open script canceled the Open operation mFV_CancelFileIsText: the file was text, so the user or the Open script canceled the Open operation mFV_CancelFileIsInUse: the file was in use, so the user or the Open script canceled the Open operation mFV_CancelFileHasStructure: the file had structure, so the user or the script canceled the Open operation mFV_CancelReferencedFilesNotFound: the file contained referenced files that were not available, so the user or the Open script canceled the Open operation mFV_CancelLanguagesNotFound: the file contained languages that weren't available, so the user or the Open script canceled the Open operation mFV_CancelFontsMapped: the document contained fonts that needed to be mapped to other fonts, so the user or the Open script canceled the Open operation mFV_CancelFontMetricsChanged: the file contained fonts with changed metrics, so the user or the Open script canceled the Open operation mFV_CancelFontsMappedInCatalog: the document's Character Catalog or Paragraph Catalog contained fonts that needed to be mapped to other fonts, so the user or the Open script canceled the Open operation

mFS_OpenNativeError and FA_errno values

Possible mFS_OpenStatus flags

`mFV_CancelFileIsDoc`: the file was a document and the Open script disallowed it

`mFV_CancelFileIsMIF`: the file was a MIF file and the Open script disallowed it

`mFV_CancelBook`: the file was a book and the Open script disallowed it

`mFV_CancelBookMIF`: the file was a MIF file and the Open script disallowed it

`mFV_CancelFileIsFilterable`: the file was a filterable file and the Open script disallowed it

`mFV_CancelFileIsOldVersion`: the file was from an old version of a FrameMaker product, so the user or the Open script canceled the Open operation

`mFV_CancelFileIsSgml`: the file was an SGML document and the Open script disallowed it

`mFV_CancelFileIsXml`: the file was an XML document and the Open script disallowed it

`mFV_UserCanceled`: the user canceled the Open operation

`mFV_CancelFileBrowser`: the user canceled the Open operation from the file browser

`mFV_CancelTempDiskFull`: there was insufficient room on the disk to cache data while opening the file.

Examples

The following code sets various open parameters and opens a book.

```
Dim oBook As FMBook


openParams = mySession.GetOpenDefaultParams
For i = 0 To (UBound(openParams) - 1)
     Select Case openParams(i).propIdent.num
          Case Is = mFS_FileIsOldVersion
               openParams(i).propVal.Value = mFV_DoOK
          Case Is = mFS_FileIsStructured
               openParams(i).propVal.Value = mFV_DoShowDialog
          Case Is = mFS_FontChangedMetric
               openParams(i).propVal.Value = mFV_DoShowDialog
```

```
            Case Is = mFS_FontNotFoundInCatalog
                 openParams(i).propVal.Value = mFV_DoShowDialog
            Case Is = mFS_FontNotFoundInDoc
                 openParams(i).propVal.Value = mFV_DoShowDialog
            Case Is = mFS_LanguageNotAvailable
                 openParams(i).propVal.Value = mFV_DoShowDialog
            Case Is = mFS_RefFileNotFound
                 openParams(i).propVal.Value = mFV_DoShowDialog
            Case Is = mFS_UseAutoSaveFile
                 openParams(i).propVal.Value = mFV_DoShowDialog
            Case Is = mFS_UseRecoverFile
                 openParams(i).propVal.Value = mFV_DoShowDialog
        End Select
    Next i
    Set oBook = mySession.OpenDocument("C:\temp\myBook.book", openParams, returnParams)
```

OpenDocument

Opens a document or creates a new document.

Synopsis

```
OpenDocument(filename As String,
OpenParams,
OpenReturnParams) As FMDocument
```

Arguments

filename The full pathname of the document file to open. If you are creating a new, untitled document, specify the filename of the template to use.

OpenParams A property list telling FrameAC how to open the book file and how to respond to errors and other conditions. To use the default list, specify `Nothing`.

OpenReturnParams A property list that returns the filename and provides information

Details

To get a property list for `OpenParams`, use the GetOpenDefaultParams method. For a list of all the properties and values, see the documentation for that method. To create a new untitled document, set the `mFS_NewDoc` property to `True` before you use the OpenDocument method.

Returns

The FMBook object that the OpenBook method creates, or `Nothing` on failure.

The `OpenReturnParams` list has the following properties, which indicate the status of the operation:

Property	Meaning and Possible Values
`mFS_OpenedFileName`	A string that specifies the opened file's pathname. If you scripted `mFS_ShowBrowser`, or the file was filtered, or you didn't specify the pathname, this pathname can be different from the one you specified in the Open script.
`mFS_OpenNativeError`	The error condition; normally the same value as `FA_errno`. If the file is opened successfully, it is set to `mFE_Success`. See the table below for a list of possible values.
`mFS_OpenStatus`	A bit field indicating what happened when the file was opened. See the table below for a list of possible flags.

Both the mFS_OpenNativeError property and the FA_errno session property indicate the result of a call to the Open method. The following table lists the possible status flags and the FA_errno and mFS_OpenNativeError values associated with them:

mFS_OpenNativeError and FA_errno values	Possible mFS_OpenStatus flags
mFE_BadParameter (file wasn't opened)	mFV_FileHadStructure: file had FrameMaker features, but current FrameMaker product is not FrameMaker
	mFV_FileAlreadyOpenThisSession: file is already open and script disallowed opening another copy
	mFV_BadFileType: file was an executable file or other unreadable type
	mFV_BadFileName: specified filename was invalid
	mFV_CantNewBooks: script specified a book that didn't exist (the Open operation can't create a new book)
	mFV_BadScriptValue: Open script contained an invalid property value
	mFV_MissingScript: OpenBook called without a script
	mFV_CantForceOpenAsText: Open script attempted to open the file as text, but file was wrong type
	mFV_DisallowedType: file was a Frame binary document and the Open script disallowed it
	mFV_DocDamagedByTextFilter: file was a text document and was damaged when it was filtered
	mFV_DocHeadersDamaged: the document headers were damaged (probably because of a file system problem)
	mFV_DocWrongSize: file is the wrong size (probably because of a file system problem)
	mFV_ChecksumDamage: bad checksum
mFE_SystemError (file wasn't opened)	mFV_TooManyWindows: too many windows were open
	mFV_BadTemplate: a bad template was specified
	mFV_FileNotReadable: don't have read permission for the file
mFE_Success (file was opened)	mFV_FileHasNewName: filename was changed from the name specified in the F_ApiOpen() call
	mFV_RecoverFileUsed: recover file was present, and it was used
	mFV_AutoSaveFileUsed: Autosave file was present, and the user or the Open script chose to use it
	mFV_FileWasFiltered: file was filterable and it was filtered

mFS_OpenNativeError and FA_errno values	Possible mFS_OpenStatus flags
	mFV_FontsWereMapped: the document contained unavailable fonts, which were mapped to substitute fonts
	mFV_FontMetricsChanged: the file contained fonts with changed metrics, but it was opened anyway
	mFV_FontsMappedInCatalog: the Paragraph or Character Catalog used unavailable fonts, which were mapped to substitute fonts
	mFV_LanguagesWerentFound: the document used some unavailable languages, but it was opened anyway
	mFV_BeefyDoc: the document file was extremely large (Macintosh platforms only), but it was opened anyway
	mFV_FileIsOldVersion: the file was from an old FrameMaker product version, but the user or the Open script chose to open it anyway
	mFV_FileStructureStripped: the file had FrameMaker features, which the user or the Open script chose to strip
	mFV_FileIsText: the file was a Text Only file, but the user or the Open script chose to open it anyway
	mFV_OpenedViewOnly: the user or the Open script chose to open the file as a View Only file
	mFV_EditableCopyOpened: the file was in use and the user or the Open script opened an editable copy
	mFV_BadFileRefsWereMapped: file reference contained illegal characters; the illegal characters were converted to something safe for the current platform
	mFV_ReferencedFilesWerentFound: imported graphics files couldn't be found, but the file was opened anyway
	mFV_FileAlreadyOpen: the file was in use and the user or the Open script opened another copy (Macintosh platforms only)
	mFV_UnresolvedXRefs: there were unresolved cross-references, but the file was opened anyway
	mFV_UnresolvedTextInsets: there were unresolved text insets, but the file was opened anyway

mFS_OpenNativeError and FA_errno values	Possible mFS_OpenStatus flags
mFE_Success, mFE_Canceled, mFE_FailedState, or mFE_CanceledByClient	mFV_LockWasReset: file lock was reset mFV_LockNotReset: file had lock that wasn't reset mFV_LockCouldntBeReset: file had lock that couldn't be reset mFV_FileWasInUse: file was in use mFV_FileIsViewOnly: file is a View Only file mFV_LockWasInvalid: file had invalid lock mFV_FileIsNotWritable: The file was not writable, and the user canceled the open via the alert. mFV_FileModDateChanged: The file has changed since the last time it was opened or saved in the current session.
mFE_Canceled (file wasn't opened)	mFV_CancelUseRecoverFile: a recover file was present, so the user or the Open script canceled the Open operation mFV_CancelUseAutoSaveFile: an Autosave file was present, so the user or the Open script canceled the Open operation mFV_CancelFileIsText: the file was text, so the user or the Open script canceled the Open operation mFV_CancelFileIsInUse: the file was in use, so the user or the Open script canceled the Open operation mFV_CancelFileHasStructure: the file had structure, so the user or the script canceled the Open operation mFV_CancelReferencedFilesNotFound: the file contained referenced files that were not available, so the user or the Open script canceled the Open operation mFV_CancelLanguagesNotFound: the file contained languages that weren't available, so the user or the Open script canceled the Open operation mFV_CancelFontsMapped: the document contained fonts that needed to be mapped to other fonts, so the user or the Open script canceled the Open operation mFV_CancelFontMetricsChanged: the file contained fonts with changed metrics, so the user or the Open script canceled the Open operation mFV_CancelFontsMappedInCatalog: the document's Character Catalog or Paragraph Catalog contained fonts that needed to be mapped to other fonts, so the user or the Open script canceled the Open operation

mFS_OpenNativeError and FA_errno values

Possible mFS_OpenStatus flags

`mFV_CancelFileIsDoc`: the file was a document and the Open script disallowed it

`mFV_CancelFileIsMIF`: the file was a MIF file and the Open script disallowed it

`mFV_CancelBook`: the file was a book and the Open script disallowed it

`mFV_CancelBookMIF`: the file was a MIF file and the Open script disallowed it

`mFV_CancelFileIsFilterable`: the file was a filterable file and the Open script disallowed it

`mFV_CancelFileIsOldVersion`: the file was from an old version of a FrameMaker product, so the user or the Open script canceled the Open operation

`mFV_CancelFileIsSgml`: the file was an SGML document and the Open script disallowed it

`mFV_CancelFileIsXml`: the file was an XML document and the Open script disallowed it

`mFV_UserCanceled`: the user canceled the Open operation

`mFV_CancelFileBrowser`: the user canceled the Open operation from the file browser

`mFV_CancelTempDiskFull`: there was insufficient room on the disk to cache data while opening the file.

Examples

The following code sets various open parameters and opens a document.

```
Dim oDoc As FMDocument


openParams = mySession.GetOpenDefaultParams
For i = 0 To (UBound(openParams) - 1)
     Select Case openParams(i).propIdent.num
          Case Is = mFS_FileIsOldVersion
               openParams(i).propVal.Value = mFV_DoOK
          Case Is = mFS_FileIsStructured
               openParams(i).propVal.Value = mFV_DoShowDialog
          Case Is = mFS_FontChangedMetric
               openParams(i).propVal.Value = mFV_DoShowDialog
```

```
            Case Is = mFS_FontNotFoundInCatalog
                 openParams(i).propVal.Value = mFV_DoShowDialog
            Case Is = mFS_FontNotFoundInDoc
                 openParams(i).propVal.Value = mFV_DoShowDialog
            Case Is = mFS_LanguageNotAvailable
                 openParams(i).propVal.Value = mFV_DoShowDialog
            Case Is = mFS_RefFileNotFound
                 openParams(i).propVal.Value = mFV_DoShowDialog
            Case Is = mFS_UseAutoSaveFile
                 openParams(i).propVal.Value = mFV_DoShowDialog
            Case Is = mFS_UseRecoverFile
                 openParams(i).propVal.Value = mFV_DoShowDialog
       End Select
  Next i
  Set oDoc = mySession.OpenDocument("C:\temp\myDoc.fm", openParams, returnParams)
```

Paste

Pastes the contents of the FrameMaker product Clipboard into the current document at the insertion point. Cutting and Pasting objects will cause FrameMaker to create a new UID for the pasted object.

Synopsis

```
Paste(flags As Long) As Long
```

Arguments

Flags	Binary flags that specify how to paste the clipboard contents, and how to handle interactive alerts. For default settings, use `0`.

Details

If you specify `0` for `flags`, this method suppresses any interactive alerts or warnings that may arise. It also inserts columns to the left of current columns and rows above the current row when pasting into tables.

You can OR the following values into `flags`:

flags constant	Meaning
mFF_INTERACTIVE	Prompt user with dialog or alert boxes that arise.
mFF_VISIBLE_ONLY	Cut only the visible portion of the selection.
mFF_DONT_DELETE_HIDDEN_TEXT	Don't replace hidden text.

flags constant	Meaning
mFF_DONT_APPLY_ALL_ROWS	Don't apply condition setting on the Clipboard to all rows. If whole table is selected and the Clipboard contains condition setting, cancel the paste.
mFF_REPLACE_CELLS	Replace selected cells with cells on the Clipboard.
mFF_INSERT_BELOW_RIGHT	Add columns to the right of the current column or below the current row.

When you use this method to paste table cells into a table, it does not work exactly like the interactive Paste command. The interactive Paste command automatically overwrites cells if the Clipboard contains less than an entire row or column. For example, if the insertion point is in a three-column table and the Clipboard contains a single cell, the interactive Paste command overwrites the cell containing the insertion point with the cell on the Clipboard. If two cells in the table are selected, the Paste command overwrites both of them with the cell on the Clipboard.

By default, the Paste method does not overwrite any cells. If the Clipboard contains less than an entire row or column when you call this method, or if the current selection is less than an entire row, The Paste method does nothing and returns FE_BadSelectionForOperation. This ensures that your do not inadvertently overwrite any cells. To make the Paste method replace cells with the Clipboard contents, you must call it with the mFF_REPLACE_CELLS flag set.

The mFF_INTERACTIVE flag takes precedence over other flags. So, if you specify mFF_INTERACTIVE | mFF_DONT_DELETE_HIDDEN_TEXT and the selection contains hidden text, the FrameMaker product prompts the user and allows the user to choose whether to delete the hidden text. It is illegal to specify mFF_REPLACE_CELLS | mFF_INSERT_BELOW_RIGHT.

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
FE_BadOperation	The function call specified an illegal operation
mFE_BadSelectionForOperation	Current text selection is invalid for this operation
mFE_Canceled	User canceled the operation

Examples

The following code ensures the current selection is an insertion point, then pastes the clipboard contents at that location.

```
Dim f_tRange As udtTextRange
Dim oDoc As FMDocument
```

```
Set oDoc = mySession.ActiveDoc
f_tRange = oDoc.TextSelection
f_tRange.end = f_tRange.beg
oDoc.TextSelection = f_tRange
oDoc.Paste (0)
```

PopClipboard

Pops the Clipboard stack, moving the entry on the top of the stack to the Clipboard.

Synopsis

```
PopClipboard() As Long
```

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`FE_Transport`	A transport error occurred
`FE_BadOperation`	The function call specified an illegal operation

Examples

The following code pushes the current clipboard onto the stack, then copies the current selection. It then pastes the copied selection to the position at the beginning ot the selection, and pops the stack data back to the clipboard.

```
im f_tRange As udtTextRange
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
f_tRange = oDoc.TextSelection
oDoc.PushClipboard
oDoc.Copy (0)
f_tRange.end = f_tRange.beg
oDoc.TextSelection = f_tRange
oDoc.Paste (0)
oDoc.PopClipboard
```

5

Print

Prints a document or a book using the default print settings. Default print settings are the settings that appear in the Print dialog box when the user attempts to print a document. This method initializes the print page size and printer name if they don't have values.

Synopsis

```
Print() As Long
```

Details

This method prints a document or book using the print setup as it is for the current session and file. To change the print settings, set the document's print properties (see "Print Properties" on page 108). To change the printer, you interact with the printer installations in the Windows OS. For more information, see your Windows programming documentation.

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`FE_BadParameter`	The function call specified an invalid parameter
`mFE_SystemError`	Couldn't open or close the printer file

Examples

The following code checks the page numbering style for the active document. If the numbering is numeric, the code sets a range of pages to print, then prints those pages to a file.

```
Dim oDoc As FMDocument


Set oDoc = mySession.ActiveDoc
If oDoc.PageNumStyle = mFV_PAGE_NUM_NUMERIC Then
     oDoc.PrintScope = mFV_PR_RANGE
     oDoc.PrintStartPage = 1
     oDoc.PrintEndPage = 2
     oDoc.PrintFileName = "C:\tmp\printedFile.ps"
     oDoc.PrintToFile = True
     oDoc.Print
End If
```

PromoteElement

Promotes the selected structural element or elements. The selected element becomes a sibling of its former parent. It appears immediately after its former parent. The siblings that follow it become its children.

Synopsis

```
PromoteElement() As Long
```

Details

This is a method of the current FMDocument object. To use this method, you must have at least one structural element completely selected.

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_BadSelectionForOperation`	Current text selection is invalid for this operation

Examples

The following code promotes the currently selected element.

```
Set oDoc = mySession.ActiveDoc
oDoc.ProElement
```

PushClipboard

Pushes the current clipboard contents onto the top of the clipboard stack.

Synopsis

```
PushClipboard() As Long
```

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`FE_Transport`	A transport error occurred

Examples

The following code pushes the current clipboard onto the stack, then copies the current selection. It then pastes the copied selection to the position at the beginning ot the selection, and pops the stack data back to the clipboard.

```
im f_tRange As udtTextRange
Dim oDoc As FMDocument
```

```
Set oDoc = mySession.ActiveDoc
f_tRange = oDoc.TextSelection
oDoc.PushClipboard
oDoc.Copy (0)
f_tRange.end = f_tRange.beg
oDoc.TextSelection = f_tRange
oDoc.Paste (0)
oDoc.PopClipboard
```

Redisplay

Updates the display for a specific document to reflect any changes that occurred while the session's `Displaying` property was set to `False`. If you have set `Displaying` to `False` and subsequently reset it to `True`, you should call this method to redisplay each document you modified.

Synopsis

```
Redisplay() As Long
```

Returns

`mFE_Success` if it succeeds,:

Examples

The following code turns off displaying for the session, then turns it on again. Then it loops through all open documents in the session and redisplays them..

```
Dim oDoc As FMDocument

mySession.Displaying = False
'*****************************************
' Perform display-intensive processing here.
' Then reset displaying & redisplay all docs.
'*****************************************
mySession.Displaying = True
Set oDoc = mySession.FirstOpenDoc
Do Until oDoc Is Nothing
     oDoc.Redisplay
     Set oDoc = oDoc.NextOpenDocInSession
Loop
```

Reformat

Reformats the specified document. If you have disabled and subsequently re-enabled reformatting by setting the session Reformatting property you should then call this method to reformat each changed document in the session.

Synopsis

```
Reformat() As Long
```

Returns

`mFE_Success` if it succeeds,:

Examples

The following code turns off displaying for the session, then turns it on again. Then it loops through all open documents in the session and redisplays them..

```
Dim oDoc As FMDocument


mySession.Reformatting = False
'****************************************
' Perform format-intensive processing here.
' Then reset formatting & reformat all docs.
'****************************************
mySession.Reformatting = True
Set oDoc = mySession.FirstOpenDoc
Do Until oDoc Is Nothing
     oDoc.Reformat
     Set oDoc = oDoc.NextOpenDocInSession
Loop
```

Rehyphenate

Rehyphenates a specified document based on changes the user has made to words' hyphenation points.

Synopsis

```
Rehyphenate() As Long
```

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_WrongProduct	Current FrameMaker product doesn't support this operation
mFE_SystemError	Couldn't allocate memory

Examples

The following code rehyphenates the active document.

```
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
oDoc.Rehyphenate
```

ResetEqnSettings

Resets the document equation settings to the default settings.

Synopsis

```
ResetEqnSettings() As Long
```

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_WrongProduct	Current FrameMaker product doesn't support this operation
mFE_SystemError	Couldn't allocate memory

Examples

The following code resets the the equation settings in the active document.

```
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
oDoc.ResetEqnSettings
```

ResetReferenceFrames

Resets the reference frames in the specific document. It is useful for updating a document after you have programmatically changed a reference frame that is referenced by paragraphs in the document.

Synopsis

```
ResetReferenceFrames() As Long
```

Arguments

Arg Name. Arg Val

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_WrongProduct	Current FrameMaker product doesn't support this operation
mFE_SystemError	Couldn't allocate memory

Examples

The following code resets the reference frames in the active document.

```
Dim oDoc As FMDocument


Set oDoc = mySession.ActiveDoc
oDoc.ResetReferenceFrames
```

RestartPgfNumbering

Restarts the paragraph numbering for a specified document. For more information on paragraph numbering, see your FrameMaker product user documentation.

Synopsis

```
RestartPgfNumbering() As Long
```

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_WrongProduct	Current FrameMaker product doesn't support this operation
mFE_SystemError	Couldn't allocate memory

Examples

The following code restarts paragraph numbering for the active document.

```
Dim oDoc As FMDocument


Set oDoc = mySession.ActiveDoc
oDoc.RestartPgfNumbering
```

5

Save

Saves a document or book. It allows you to script the way the FrameMaker product saves the file and to specify responses to warnings and messages that arise while the file is being saved. You can save a file under its current name or save it as a new file.

Synopsis

```
Save(filename As String,
SaveParameters,
SaveReturnParameters) As FMDocument
```

Arguments

filename The pathname for saving the document or book

SaveParameters property list that tells the FrameMaker product how to save the file and how to respond to errors and other conditions. Use the GetSaveDefaultParams method to get this property list. To use the default list, specify `Nothing`.

SaveReturnParameters A property list that returns information about how the FrameMaker product saved the file

Details

To get a property list for `SaveParameters`, you can use GetSaveDefaultParams and modify individual properties in the list it returns. See the documentation for that method for a list of the parameters it returns.

The SaveReturns^Parameters is a list of properties that indicate the status of the Save operation. This list can have the properties and values shown in the following table:

Property	Meaning and possible values
`mFS_SavedFileName`	A string that specifies the saved file's full pathname.
`mFS_SaveNativeError`	The error condition. If the file is saved successfully, it is set to `mFE_Success`. See the table below for the possible values.
`mFS_SaveStatus`	A bit field indicating what happened when the file was saved. See the table below for the possible values.

Both the `mFS_SaveNativeError` property and the `FA_errno` session property indicate the result of a call to the Save method. The `mFS_SaveStatus` flags indicate how or why this result occurred. The following table lists the possible status flags and the `FA_errno` and `mFS_SaveNativeError` values associated with them.

Property	Meaning and possible values
`mFE_Success`	None.

Property	Meaning and possible values
mFE_Canceled or FmE_CanceledByClient (file wasn't saved)	mFV_FileNotWritable: file was not writable.
	mFV_BadSaveFileName: specified file name is not allowed by the operating system.
	mFV_BadFileId: the file's operating system ID was bad.
	mFV_BadSaveScriptValue: script specified by SaveParameters had an invalid value.
	mFV_CancelSaveFileIsInUse: The file is in use and the user did not or could not reset the lock. Or the file is in use, and the mFS_FileIsInUse script is set to mFV_DoCancel, or it is set to mFV_ResetLockAndContinue but the FrameMaker product could not reset the lock.
	mFV_CancelSaveModDateChanged: The file has changed since the last time it was opened or saved in the current session.
	mFV_FileWasInUse: file was in use.
	mFV_LockCouldntBeReset: file lock couldn't be reset.
	mFV_LockWasReset: file lock was reset.
	mFV_LockNotReset: file lock was not reset.
	mFV_FileIsViewOnly: file was View Only.
mFE_WrongProduct	No associated values. The specified file contains structure and and the current product interface is not Structured FrameMaker.
mFE_FailedState or mFE_BadParameter	No associated values. The filename was invalid.
mFE_FilterFailed	mFV_InvalidSaveFilter: The filter specified by mFS_SaveFileTypeHint is not installed, or the syntax for mFS_SaveFileTypeHint is invalid.

Returns

The FMDocument or FMBook on success, or Nothing on failure.

Examples

See the example for the GetSaveDefaultParams method.

ScrollToText

Scrolls the document window to a specified text range. It scrolls to the end of the range that is closest to the current display position.

Synopsis

```
ScrollToText(textRange) As Long
```

Arguments

textRange	A udtTextRange that expresses the text range to scroll to

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_NotTextObject	One or both of the IDs specified by textRange is not the ID of an object that contains text, such as a paragraph (FMParagraph) or a flow (FMFlow)
mFE_OffsetNotFound	Offset specified for the text range couldn't be found in the specified paragraph or text line
mFE_BadRange	Specified text range is invalid

Examples

The following code scrolls the document to the insertion point or to the end of the current text selection. It then redisplays the document to make sure it displays the new location.

```
Dim f_tRange As udtTextRange
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
f_tRange = oDoc.TextSelection
oDoc.ScrollToText f_tRange
oDoc.Redisplay
```

SetNotification

Requests that the FrameMaker product notify your client whenever a specified event, or stage of an event, occurs.

Synopsis

```
SetNotification(notification As Long,
state As Long)
```

Arguments

notification	Constant that specifies the notification point — see the following table for a list of constants
state	The notification state — True to turn it on, and False to turn it off

Details

The notifications you specify work in conjunction with the OnApiNotify event — for every notification point you turn on in your application or plug-in, FrameMaker sends that application or plug-in a notification for the corresponding event. In this way, you can trap specific user actions, or actions performed by FrameMaker or other plug-ins, and modify the associated processes.

> **Important:** If the FrameMaker product encounters an internal error and exits, it does not send any notification to your application or plug-in about operations performed after the error occured. For example, assume an error causes FrameMaker to quit. The product allows the user to save changes in open documents, but it does not notify any plug-ins of those save operations.

Many actions have several notification points or stages for which you can request notification. The following table lists the notification points, and the constants that specify them:

Event or operation	Notification points	Notification constants
Frame binary document opened	Before checking the type of the file to be opened	`mFA_Note_PreFileType`
	After checking the type of the file to be opened	`mFA_Note_PostFileType`
	Before opening the file	`mFA_Note_PreOpenDoc`
	After opening the file	`mFA_Note_PostOpenDoc`
MIF document opened	Before checking the type of the file to be opened	`mFA_Note_PreFileType`
	After checking the type of the file to be opened	`mFA_Note_PostFileType`
	Before opening the file	`mFA_Note_PreOpenMIF`
	After opening the file	`mFA_Note_PostOpenMIF`
SGML document opened	Before checking the type of the file to be opened	`mFA_Note_PreFileType`
	After checking the type of the file to be opened	`mFA_Note_PostFileType`
	Before opening the file	`mFA_Note_PreOpenSGML`
	After opening the file	`mFA_Note_PostOpenSGML`

Event or operation	Notification points	Notification constants
XML document opened	Before checking the type of the file to be opened	mFA_Note_PreFileType
	After checking the type of the file to be opened	mFA_Note_PostFileType
	Before opening the file	mFA_Note_PreOpenXML
	After opening the file	mFA_Note_PostOpenXML
Filterable document opened	Before checking the type of the file to be opened	mFA_Note_FilterIn
Frame binary book opened	Before checking the type of the file to be opened	mFA_Note_PreFileType
	After checking the type of the file to be opened	mFA_Note_PostFileType
	Before opening the file	mFA_Note_PreOpenBook
	After opening the file	mFA_Note_PostOpenBook
MIF book opened	Before checking the type of the file to be opened	mFA_Note_PreFileType
	After checking the type of the file to be opened	mFA_Note_PostFileType
	Before opening the file	mFA_Note_PreOpenBookMIF
	After opening the file	mFA_Note_PostOpenBookMIF
User double-clicked to open a document in a book window	Before opening the file	mFA_Note_PreBookComponentOpen
	After opening the file	mFA_Note_PostBookComponentOpen
Generating a list or TOC for a document or a book	Before generating the file	mFA_Note_PreGenerate
	After generating the file	mFA_Note_PostGenerate
Document saved in Frame binary format	Before saving the document	mFA_Note_PreSaveDoc
	After saving the document	mFA_Note_PostSaveDoc
Document saved as MIF	Before saving the file as MIF	mFA_Note_PreSaveSGML
	After saving the file as MIF	mFA_Note_PostSaveSGML

Event or operation	Notification points	Notification constants
Document saved as SGML	Before saving the file as SGML	mFA_Note_PreSaveSGML
	After saving the file as SGML	mFA_Note_PostSaveSGML
Document saved as XML	Before saving the file as XML	mFA_Note_PreSaveSGML
	After saving the file as XML	mFA_Note_PostSaveSGML
Document saved as PDF	Before specifying Acrobat settings and generating PostScript	mFA_Note_PreSaveAsPDFDialog
	After specifying Acrobat settings and generating PostScript	mFA_Note_PostSaveAsPDFDialog
	Before distilling the PostScript	mFA_Note_PreDistill
	After distilling the PostScript	mFA_Note_PostDistill
Document saved as filterable type	Before the document is saved	mFA_Note_FilterOut
Document exited	Before exiting the document	mFA_Note_PreQuitDoc
	After exiting the document	mFA_Note_PostQuitDoc
Book exited	Before exiting the book	mFA_Note_PreQuitBook
	After exiting the book	mFA_Note_PostQuitBook
First change made to a document since it was opened or saved	After the document is changed	mFA_Note_DirtyDoc
First change made to a book since it was opened or saved	After the book is changed	mFA_Note_DirtyBook
Book saved in Frame binary format	Before saving the book	mFA_Note_PreSaveBook
	After saving the book	mFA_Note_PostSaveBook
Book saved in MIF format	Before saving the MIF file	mFA_Note_PreSaveBookMIF
	After saving the MIF file	mFA_Note_PostSaveBookMIF

Event or operation	Notification points	Notification constants
Document saved with Autosave	Before saving the document	mFA_Note_PreAutoSaveDoc
	After saving the document	mFA_Note_PostAutoSaveDoc
Document reverted	Before reverting the document	mFA_Note_PreRevertDoc
	After reverting the document	mFA_Note_PostRevertDoc
Book reverted	Before reverting the book	mFA_Note_PreRevertBook
	After reverting the book	mFA_Note_PostRevertBook
FrameMaker product exited	Before the OK to Exit dialog box appears	mFA_Note_PreQuitSession
	Immediately before exiting the session	mFA_Note_PostQuitSession
Another client calls F_ApiCallClient() with clname set to the current client's name	After the call has been made to F_ApiCallClient()	mFA_Note_ClientCall
Any user action, such as a command choice or text entry	After the FrameMaker product finishes processing the user action	mFA_Note_BackToUser
Text inset owned by current client clicked	After the user clicked the inset	mFA_Note_DisplayClientTiDialog
FrameMaker product updating all text insets	When the client needs to update insets that belong to it	mFA_Note_UpdateAllClientTi
FrameMaker product updating a specific text inset	When the client needs to update a specified inset	mFA_Note_UpdateClientTi
Text or graphic imported	Before importing the text or graphic	mFA_Note_PreImport
	After importing the text or graphic	mFA_Note_PostImport

Event or operation	Notification points	Notification constants
FrameMaker product command invoked or text entered in a document	Before the FrameMaker product executes command or adds text to the document	`mFA_Note_PreFunction`
	After the FrameMaker product executes command or adds text to the document	`mFA_Note_PostFunction`
Mouse button clicked	Before the FrameMaker product responds to the mouse click	`mFA_Note_PreMouseCommand`
	After the FrameMaker product responds to the mouse click	`mFA_Note_PostMouseCommand`
Hypertext command invoked	Before the FrameMaker product executes the hypertext command	`mFA_Note_PreHypertext`
	After the FrameMaker product executes the hypertext command	`mFA_Note_PostHypertext`
The user clicked Go To Source in the cross-reference dialog box	Before the FrameMaker product goes to the cross-reference source	`mFA_Note_PreGoToXrefSrc`
	After the FrameMaker product goes to the cross-reference source	`mFA_Note_PostGoToXrefSrc`
Document or book printed	After the user clicks OK in the Print dialog box, but before the FrameMaker product prints the document or book	`mFA_Note_PrePrint`
	After the FrameMaker product prints the document or book	`mFA_Note_PostPrint`
Body page added to document	After the FrameMaker product adds the body page	`mFA_Note_BodyPageAdded`

Event or operation	Notification points	Notification constants
Body page deleted from document	After the FrameMaker product deletes the body page	`mFA_Note_BodyPageDeleted`
Structural element inserted	Before the element is inserted	`mFA_Note_PreInsertElement`
Notifications issued when user inserts an element, marker, footnote, cross-reference, equation, anchored frame, or table. Not issued when FrameMaker automatically inserts elements for added rows, columns, or table titles; imported graphics; or when a plug-in adds an element	After the element is inserted	`mFA_Note_PostInsertElement`
Structural element copied	Before the element is copied.	`mFA_Note_PreCopyElement`
	After the element is copied.	`mFA_Note_PostCopyElement`
Structural element changed	Before the element is changed	`mFA_Note_PreChangeElement`
	After the element is changed	`mFA_Note_PostChangeElement`
Structural element wrapped	Before the element is wrapped	`mFA_Note_PreWrapElement`
	After the element is wrapped	`mFA_Note_PostWrapElement`
Structural element dragged	Before the element is dragged	`mFA_Note_PreDragElement`
	After the element is dragged	`mFA_Note_PostDragElement`
An attribute value is set	Before the attribute value is set	`mFA_Note_PreSetAttrValue`
	After the attribute value is set	`mFA_Note_PostSetAttrValue`

Event or operation	Notification points	Notification constants
Element definitions are imported	Before the element definitions are imported	mFA_Note_PreImportElemDefs
	After the element definitions are imported	mFA_Note_PostImportElemDefs
Inline input of double-byte text	Before the text entry	mFA_Note_PreInlineTypeIn
	After the text entry	mFA_Note_PostInlineTypeIn
Filter a file on import or export via a filt to file filter	A file to file filter has been invoked—this notification occurs before the file is imported.	mFA_Note_FilterFileToFile

These notification constants are numbered sequentially, starting from zero. FrameAC provides a constant, mFA_Note_Num, that specifies the total number of notification constants. With this value, you can easily set up a loop to request notification for all notification points, as shown in the example code below.

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_Transport	A transport error occurred
mFE_BadNotificationNum	The specified notification number was invalid

Examples

For an example of how to respond to the notifications you request, see the documentation for the OnApiNotify event. The following code shows how to loop and request every notification.

```
Dim noteNum As Long


noteNum = 0
For noteNum = 0 To mFA_Note_Num - 1
     mySession.SetNotification noteNum, True
Next noteNum
```

SetPropVals

Applies a list of properties and values to an object. You can use this method and GetPropVals to copy formats from one object to another. For example, you can copy properties from one graphic object to another, from a paragraph format to a paragraph, or from one paragraph to another paragraph

Synopsis

```
SetPropVals(Properties) As Long
```

Arguments

Properties The property list

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_BadName	Specified name is illegal
mFE_BadNewFrame	FrameAC can't move the specified object to this frame
mFE_BadNewGroup	FrameAC can't move the specified object to this graphic object group (FMGroup)
mFE_BadNewSibling	Object can't be made a sibling of the specified object
mFE_BadObjId	Invalid object ID
mFE_BadPropNum	Specified property number is invalid
mFE_BadPropType	Incorrect property type for this function
mFE_BadRange	Specified text range is invalid
mFE_CantSmooth	Object can't be smoothed
mFE_DupName	Property can't be set to this name because it is already used by another object
mFE_GenRuleAmbiguous	General rule in structured document was ambiguous
mFE_GenRuleConnectorExpected	General rule in structured document missing connector
mFE_GenRuleItemExpected	General rule in structured document missing rule item
mFE_GenRuleLeftBracketExpected	General rule in structured document missing left bracket
mFE_GenRuleMixedConnectors	General rule in structured document has mixed connectors
mFE_GenRuleRightBracketExpected	General rule in structured document missing right bracket
mFE_GenRuleSyntaxError	General rule in structured document has syntax error
mFE_GroupSelect	FrameAC can't select or deselect an object in the specified group
mFE_HiddenPage	Value must specify a hidden page (FMHiddenPage)
mFE_InvContextSpec	FrameAC encountered an invalid context specification in a structured FrameMaker document
mFE_NotBookComponent	Value must specify a book component (FMBookComponent)
mFE_NotFrame	Value must specify a frame
mFE_NotGraphic	Value must speciify a graphic object

Error code	Meaning
mFE_NotGroup	Value must specify a graphic object group (FMGroup)
mFE_NotTextFrame	Value must specify a text column (FMTextFrame)
mFE_NotTextObject	Object must be an object that contains text, such as a paragraph (FMParagraph) or a flow (FMFlow)
mFE_OffsetNotFound	Offset specified for the text location couldn't be found in the specified paragraph or text line
mFE_OutOfRange	Specified property value is out of the legal range for the specified property
mFE_PageFrame	Value must specify a page frame object (FMPageFrame)
mFE_ReadOnly	Property is read-only
mFE_WithinFrame	Object must be moved to a different frame first
mFE_WrongProduct	Current FrameMaker product doesn't support the operation

Examples

See the example for GetPropVals.

SetTextProps

Sets the text properties (such as the format tag, font family, and size) for a text range.

Synopsis

```
SetTextProps(textRange,
Properties) As Long
```

Arguments

textRange	A udtTextRange that expresses the text range to set properties to
Properties	The property list

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_BadName	Specified name is illegal
mFE_BadRange	Specified text range is invalid
mFE_GenRuleAmbiguous	General rule in structured document was ambiguous
mFE_GenRuleConnectorExpected	General rule in structured document missing connector
mFE_GenRuleItemExpected	General rule in structured document missing rule item

Error code	Meaning
mFE_GenRuleLeftBracketExpected	General rule in structured document missing left bracket
mFE_GenRuleMixedConnectors	General rule in structured document has mixed connectors
mFE_GenRuleRightBracketExpected	General rule in structured document missing right bracket
mFE_GenRuleSyntaxError	General rule in structured document has syntax error
mFE_NotTextObject	Value must be the ID of an object that contains text, such as a paragraph (FO_Pgf) or a flow (FO_Flow)
mFE_OffsetNotFound	Offset specified for the text location couldn't be found in the specified paragraph or text line
mFE_OutOfRange	Specified property value is out of the legal range for the specified property
mFE_ReadOnly	Property is read-only
mFE_WrongProduct	Current FrameMaker product doesn't support the operation

Examples

See the example for GetTextProps.

SimpleOpenBook

Opens a book.

Synopsis

```
SimpleOpenBook(filename As String,
interactive As Long) As FMBook
```

Arguments

filename	The absolute pathname of the book to open.
interactive	Specifies whether the FrameMaker product displays messages and warnings to the user. `True` instructs the FrameMaker product to display messages and warnings

Details

If you call this method with `interactive` set to `True`, the FrameMaker product displays the Open dialog box. It uses the path specified by the session property, `mFP_OpenDir`, as the default path. If a warning or error condition arises, the FrameMaker product notifies the user.

If you set `interactive` to `False`, the FrameMaker product does not display the Open dialog box or other messages and warnings. If it is necessary to modify a file to continue opening it, this method aborts the operation without notifying the user, and returns `Nothing`.

Returns

The opened FMBook object on success or `Nothing` on failure. If this method fails, it sets an error code to `FA_errno`. The list of error codes for SimpleOpenBook is the same as for OpenBook. For the list of error codes, see the documentation for OpenBook.

Examples

The following code opens a book file.

```
Dim oBook As FMBook


Set oBook = mySession.SimpleOpenBook("C:\temp\myBook.book", False)
```

SimpleOpenDocument

Opens a book or document.

Synopsis

```
SimpleOpenDocument(filename As String,
interactive As Long) As FMDocument
```

Arguments

filename	The absolute pathname of the document to open.
interactive	Specifies whether the FrameMaker product displays messages and warnings to the user. `True` instructs the FrameMaker product to display messages and warnings

Details

If you call this method with `interactive` set to `True`, the FrameMaker product displays the Open dialog box. It uses the path specified by the session property, `mFP_OpenDir`, as the default path. If a warning or error condition arises, the FrameMaker product notifies the user.

If you set `interactive` to `False`, the FrameMaker product does not display the Open dialog box or other messages and warnings. If it is necessary to modify a file to continue opening it, this method aborts the operation without notifying the user, and returns `Nothing`.

Returns

The opened FMDocument object on success or Nothing on failure. If this method fails, it sets an error code to `FA_errno`. The list of error codes for SimpleOpenDocument is the same as for OpenDocument. For the list of error codes, see the documentation for OpenDocument.

Examples

The following code opens a document file.

```
Dim oDoc As FMDocument


Set oDoc = mySession.SimpleOpenBook("C:\temp\myDoc.fm", False)
```

SimpleSave

Saves a document or book.

Synopsis

```
SimpleSave(filename As String,
interactive As Long) As FMDocument
```

Arguments

filename	The absolute pathname for saving the document or book
interactive	Specifies whether FrameMaker displays messages and warnings to the user. `True` displays messages and warinings.

Details

If you set interactive to `False` and you specify the document or book's current name for `filename`, the FrameMaker product saves the document or book under its current name. If you specify another filename for `filename`, the FrameMaker product saves the document or book to that filename. If you specify an empty string (""), the FrameMaker product doesn't save the file. Instead it sets `FA_errno` to `mFE_BadParameter`.

If you set interactive to `True`, the FrameMaker product displays the Save dialog box and allows the user to choose a filename. The document or book's current name appears as the default name.

Returns

The FMDocument or FMBook object on success, or `Nothing` on failure.On failure, the method sets the following error code to `FA_errno`:

Error code	Meaning
`mFE_BadParameter`	The function call specified an invalid parameter

Examples

The following code opens a document, then saves it to a new name.

```
Dim oDoc As FMDocument


Set oDoc = mySession.SimpleOpenDocument("C:\tmp\Test2.fm", False)
Set oDoc = oDoc.SimpleSave("C:\tmp\Test3.fm", False)
```

SplitElement

Splits the structural element containing the insertion point into two elements at the insertion point. The insertion point must be inside the element you want to split.

Synopsis

```
SplitElement() As Long
```

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_BadSelectionForOperation`	Current text selection is invalid for this operation
`mFE_WrongProduct`	Current product interface isn't Structured FrameMaker

Examples

The following code splits the element at the current insertion point.

```
Dim oDoc As FMDocument


Set oDoc = mySession.ActiveDoc
oDoc.SplitElement
```

StraddleCells

Straddles the specified cells in a table.

Synopsis

```
StraddleCells(heightInRows As Long,
widthInCols As Long) As Long
```

Arguments

heightInRows	The number of cells to straddle vertically
widthInCOls	The number of cells to straddle horizontally

Details

The cells you straddle must all be from the same type of row. You can't straddle a set of cells that are in both heading and body rows or footing and body rows. Also, the cells you straddle must be unstraddled; you cannot use this function to further straddle cells that are already straddled.

The FMCell object from which you call this method is the first cell of the straddle — the leftmost and uppermost cell in the range of cells to straddle.

Returns

mFE_Success if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_BadParameter	The function call specified an invalid parameter
mFE_BadOperation	The function call specified an illegal operation
mFE_WrongProduct	Current FrameMaker product does not support tables

Examples

The following code straddles all the cells in the first body row of the first table in the document.

```
Dim oDoc As FMDocument
Dim oTbl As FMTable
Dim oCell As FMCell, oRow As FMRow

Set oDoc = mySession.ActiveDoc
Set oTbl = oDoc.SelectedTbl
Set oRow = oTbl.GetFirstRow(mFV_ROW_BODY)
Set oCell = oRow.FirstCellInRow
oCell.StraddleCells 1, oTbl.TblNumCols
```

TextLocToElementLoc

Returns udtElementLoc data that corresponds to the current element location (udtTextLoc).

Synopsis

```
TextLocToElementLoc(TextLocation)
```

Arguments

ElementLoc. The udtTextLoc to convert

Returns

The udtElementLoc data specifying a text location. If this method fails, it sets the following value to mFA_Errno.

Error code	Meaning
mFE_BadParameter	ElementLoc was empty, or one of its values was improperly specified
mFE_WrongProduct	Current FrameMaker product doesn't support the operation

Examples

The following code converts the beginning of the text selection to an element location, and then reports the name of the parent element.

```
Dim f_tRange As udtTextRange
Dim f_elemLoc As udtElementLoc, f_elemRange As udtElementRange
Dim oElemDef As FMElementDefinition, oElem As FMElement
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
f_tRange = oDoc.TextSelection
f_elemLoc = oDoc.TextLocToElementLoc(f_tRange.beg)
Set oElem = mySession.GetObject(f_elemLoc.parentId, oDoc)
Set oElemDef = oElem.ElementDef

mySession.Alert "Parent element: " + oElemDef.Name, mFF_ALERT_CONTINUE_NOTE
```

UnStraddleCells

Unstraddles the specified cells.

Synopsis

```
UnStraddleCells(heightInRows As Long,
widthInCols As Long) As Long
```

Arguments

heightInRows	The number of cells to unstraddle vertically
widthInCols	The number of cells to unstraddle horizontally

Details

The width and height arguments count from the straddle start (topmost and leftmost cell), starting at one.

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_BadParameter`	The function call specified an invalid parameter
`mFE_BadOperation`	The function call specified an illegal operation
`mFE_WrongProduct`	Current FrameMaker product does not support tables

Examples

The following code unstraddles the cell that contains the insertion point. The code counts the rows and columns that are straddled, using the current cell as the straddle start.

Note that the straddled cells exist in the document, but are not used when formatting and displaying the text in the table. Cells that are straddled have their CellIsStraddled property set to True. However, the first cell in a straddle has this property set to False. You can see how this example accounts for that issue.

Also note that this code only operates on body rows. Further, the code relies on a text selection within a cell to determine where to begin.

```
Dim f_tRange As udtTextRange
Dim oTbl As FMTable
Dim oCell As FMCell, strStart As FMCell
Dim oDoc As FMDocument
Dim strCols As Long, strRows As Long

Set oDoc = mySession.ActiveDoc
Set oTbl = oDoc.SelectedTbl
f_tRange = oDoc.TextSelection
' The following test would be better if it posted an alert.
If f_tRange.beg.objId = 0 Then Exit Sub
Set oPgf = mySession.GetObject(f_tRange.beg.objId, oDoc)
Set strStart = oPgf.InTextObj

' Make sure this is in a body row.
If Not mFV_ROW_BODY = strStart.CellRow.RowType Then Exit Sub

Set oCell = strStart
'Account for oCell.CellIsStraddled = False in first cell in straddle.
Set oCell = oCell.CellBelowInCol
strRows = 1
Do While Not oCell Is Nothing
     If oCell.CellIsStraddled Then
          strRows = strRows + 1
     Else
          Exit Do
     End If
```

```
     Set oCell = oCell.CellBelowInCol
Loop

Set oCell = strStart
'Account for oCell.CellIsStraddled = False in first cell in straddle.
Set oCell = oCell.NextCellInRow
strCols = 1
Do While Not oCell Is Nothing
     If oCell.CellIsStraddled Then
          strCols = strCols + 1
     Else
          Exit Do
     End If
     Set oCell = oCell.NextCellInRow
Loop
strStart.UnStraddleCells strRows, strCols
```

UnWrapElement

Removes selected structural elements, but leaves their contents and child elements intact in the document. This method does not remove all the elements in the selection, just the top-level elements.

Synopsis

```
UnWrapElement() As Long
```

Details

At least one structured element must be selected when you call this method. UnwrapElement has no effect on object elements (elements containing markers or anchored frames, for example).

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_WrongProduct`	Current product interface isn't Structured FrameMaker
`mFE_BadSelectionForOperation`	Current text selection is invalid for this operation

Examples

The following code unwraps the elements that are currently selected in the document.

```
Dim oDoc As FMDocument


Set oDoc = mySession.ActiveDoc
oDoc.UnWrapElement
```

UpdateBook

Performs Update Book commands.

Synopsis

```
UpdateBook(UpdateParams,
UpdateReturnParams) As Long
```

Arguments

UpdateParams property list that tells the FrameMaker product how to save the file and how to respond to errors and other conditions. Use the GetUpdateDefaultParams method to get this property list. To use the default list, specify `Nothing`.

UpdateReturnParams A property list that returns information about how the FrameMaker product updated the book

Details

This method allows you to specify a script (property list) telling the FrameMaker product how to update the book and how to deal with error and warning conditions. For example, you can specify whether to abort or to continue updating a book if the book contains view-only documents.

To get a property list to specify for the updateParams parameter, use GetUpdateDefaultParams. For a list of all the properties an Update Book script can include, see the documentation for that method.

While updating a book, you can post messages to the book error log. To do so, you use the CallClient method to pass your messages to the BookErrorLog client. For more information, see the FrameMaker FDK documentation

`UpdateReturnParams` is a list of properties that contains the following property:

Property	Meaning
`mFS_UpdateBookStatus`	A bit field to indicate what happened during the update. See the table below for a list of possible flags.

The following table lists the possible error codes for an update operation, plus the associated values for the FS_UpdateStatus property.

mFA_errno value	Possible mFS_UpdateBookStatus flags
mFE_BadOperation	mFV_BookNotSelfConsistent: The book is not self-consistent (book generates data in one file that is source data for another generated file, or page count continually changes for this operation)
	mFV_DuplicateFileInBook: one or more files in the book is a duplicate of another file
	mFV_NoNonGeneratedFilesInBook: the only files in the book are generated files
mFE_BadParameter	mFV_BadUpdateBookScriptValue: the update book script contained an invalid property value
mFE_Canceled mFE_CanceledByClient	mFV_CancelInconsistentNumPropsInFileInBook: one or more of the book's document files has numbering properties that are inconsistent with the properties stored in the book
	mFV_CancelNonFMFileInBook: one or more of the book's document files is not a FrameMaker product file
	mFV_CancelViewOnlyFileInBook: one or more of the book's document files is view-only
	mFV_UserCanceledUpdateBook: the user cancelled the update operation
mFE_SystemError	mFV_FileInBookNotOpened: one or more files in the book could not be opened
	mFV_FileInBookNotSaved: one or more files in the book could not be saved
	mFV_TooManyWindowsUpdateBook: too many windows were open for the currently available memory

Returns

An error code which has the same value that the method sets to FA_errno. The possible error codes mirror the values set to the UpdateReturnParams property list.

Examples

The following code gets the default update paramaters, modifies some of them, then updates the current book.

```
Dim updateParams As Variant, returnParams As Variant

Dim i As Long

Dim oBook As FMBook
```

```
Set oBook = mySession.ActiveBook
updateParams = mySession.GetUpdateBookDefaultParams
For i = 0 To (UBound(updateParams) - 1)
     Select Case updateParams(i).propIdent.num
          Case Is = mFS_AlertUserAboutFailure
               updateParams(i).propVal.Value = True
          Case Is = mFS_MakeVisible
               updateParams(i).propVal.Value = False
     End Select
Next i
oBook.UpdateBook updateParams, returnParams
```

UpdateTextInset

Updates the contents of a stale text inset. It determines whether an inset is stale by comparing the inset's `TiLastUpdate` property with the modification date of the inset's source file.

Synopsis

```
UpdateTextInset() As Long
```

Details

This method does not update a text inset unless it is stale. To make the inset stale, set the TiLastUpdate property to 0.

This method does not update graphic insets (FMInset objects)

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_BadFileType`	The inset specifies a file that does not match the import type (for example, the inset imports a binary document but the file is a text file or doesn't exist)
`mFE_SomeUnresolved`	Some text insets were unresolved
`mFE_WrongProduct`	Product doesn't support the specified operation
`mFE_SystemError`	Couldn't allocate memory

Examples

The following code updates all the text insets in a document.

```
Dim oTextInset As FMTextInset
```

```
Dim oDoc As FMDocument


Set oDoc = mySession.ActiveDoc
Set oTextInset = oDoc.FirstTiInDoc
Do Until oTextInset Is Nothing
     oTextInset.UpdateTextInset
     Set oTextInset = oTextInset.NextTiInDoc
Loop
```

UpdateVariables

Updates all the variables in a document. It performs the same operation as clicking Update in the Variable dialog box.

Synopsis

```
UpdateVariables() As Long
```

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_WrongProduct`	Product doesn't support the specified operation
`mFE_SystemError`	Couldn't allocate memory

Examples

The following code updates the variables in a document.

```
Dim oDoc As FMDocument


Set oDoc = mySession.ActiveDoc
oDoc.UpdateVariables
```

UpdateXRefs

Updates cross-references in a document. It performs the same operation as clicking Update in the Cross-Reference window.

Synopsis

```
UpdateXRefs(flags As Long) As Long
```

Arguments

flags	Flags to indicate which cross-references to update. Se the table below for values.

Details

You can OR the values listed in the following tables in the `flags` argument:

This value	For
mFF_XRUI_FORCE_UPDATE	Updates all cross-references, regardless of whether the source document has changed
mFF_XRUI_INTERNAL	Only internal cross-references
mFF_XRUI_OPEN_DOCS	Only cross-references whose sources are in open documents
mFF_XRUI_CLOSED_DOCS	Only cross-references whose sources are in closed documents
mFF_XRUI_EVERYTHING	All cross-references

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
mFE_SomeUnresolved	After the update there were unresolved references

Examples

The following code updates all the cross-references in the active document.

```
Dim oDoc As FMDocument


Set oDoc = mySession.ActiveDoc
oDoc.UpdateXRefs mFF_XRUI_EVERYTHING
```

WrapElement

Inserts a structural element around the selected text and structural elements in a document.

Synopsis

```
WrapElement(ElementDefinition As FMElementDefinition) As Long
```

Arguments

*ElementDefinition*The element definition to use for the new element

Returns

`mFE_Success` if it succeeds, or one of the following error codes:

Error code	Meaning
`mFE_WrongProduct`	Current product interface isn't Structured FrameMaker
`mFE_BadElementDefId`	Specified element definition ID is invalid
`mFE_BadSelectionForOperation`	Current text selection is invalid for this operation

Examples

The following code wraps the selected text and elements.

```
Dim oElemDef As FMElementDefinition
Dim oDoc As FMDocument

Set oDoc = mySession.ActiveDoc
Set oElemDef = mySession.GetNamedObject("MyElemDef", mFO_ElementDef, oDoc)
oDoc.WrapElement oElemDef
```

6 Data Type Reference

This section lists data objects that are specific to FrameMaker sessions, documents, and document content. Most object properties use these types of data, and many methods return these data objects, or arrays of these data objects.

F_Int

Details

F_Int objects represent most numerical values in FrameAC. They are used to represent:

- Boolean values where 1 is True and 0 is False
- Object Ids
- Constants for properties

Properties

Property:	Type:	Description:
value	Long	A numerical value

F_Metric

Details

The F_Metric value is a 32-bit fixed-point number. FrameAC uses F_Metric values to express linear measurements, such as tab offsets and font sizes. It also uses F_Metric values to express percentages and angular degrees when a high degree of accuracy is required. F_Metric values should not be confused with metric system values, that is, with centimeters and meters.

With F_Metric numbers, the 16 most significant bits represent the digits before the decimal, and the 16 least significant bits represent the digits after the decimal. Therefore, the number 1 is expressed as hexadecimal `0x10000` or decimal `65536`.

F_Metric values are used for most measurements in order to minimize rounding errors.

Using F_Metric Values for Linear Measurement

When used for linear measurement, a MetricT unit represents a point (1/72 inch). The 16 most significant bits of a MetricT value represent the digits before the decimal; the 16 least

significant bits represent the digits after the decimal. Therefore, 1 point is expressed as hexadecimal 0x10000 or decimal 65536.

The following table lists the units of the measurement systems the FrameMaker product supports, and their MetricT equivalents:

Measurement unit	F_Metric hex value	F_Metric decimal value	Relationship to other units
inch	0x480000	4718592	72 points
centimeter	0x1c58b1	1857713	2.54 cm per inch
millimeter	0x02d5ab	185771	25.4 mm per inch
pica	0x0c0000	786432	12 points
point	0x10000	65536	1/72 inch
didot	0x011159	69977	0.01483 inch
cicero	0x0cd02c	839724	12 didots

FrameAC defines the following constants which you may prefer to use:

mFV_METRIC_INCH

mFV_METRIC_CM

mFV_METRIC_MM

mFV_METRIC_PICA

mFV_METRIC_POINT

mFV_METRIC_DIDOT

mFV_METRIC_CICERO

With these constants, you can specify F_Metric values in your code in a way that is easy to understand. For example, you can specify 5 inches with `5*mFV_METRIC_INCH`.

Using F_Metric Values for Percentages and Angles

FrameAC uses F_Metric values to represent percentages. For some properties, such as a document's Zoom property and the character format's Spread, the percentage is represented as a fraction, where 1 represents 100%, and .5 represents 50%. In other words, the MetricT value 0x10000 (`65536` in decimal notation), and represents 100%.

For other properties, such as colour properties, the percentage is represented as the actual value, where 1 represents 1%, and 100 represents 100%. In this case, the F_Metric value 0x10000 (`65536` in decimal notation) represents 1%. FrameAC uses F_Metric values to represent angles in the same way, where `0x10000` represents 1 degree.

For information about F_Metric values for specific properties, see their descriptions in Chapter 3, "Object Reference."

Properties

Property:	Type:	Description:
value	long	A value representing a 32-bit fixed decimal number, where the high 16 bits represent the value to the left of the decimal point, and the low 16 bits represent the value to the right of the decimal point.

F_String

Details

The F_String object represents string data.

Properties

Property:	Type:	Description:
value	BSTR	A NULL terminated string.

F_TypedValValues

Details

The F_TypedValValues object contains data for an instance of the udtTypedVal object. The valType of the udtTypedVal object indicates which property of F_TypedValValues actually contains the data.

Properties

Property:	Type:	Description:
ssval	Array of F_String	A list of strings.
msval	Array of F_Metric	A list of F_Metric values.
psval	Array of udtPoint	A list of x, y coordinates.
tsval	Array of udtTab	A list of tab specifications.
tlval	udtTextLoc	A location in text.
trval	udtTextRange	A range of text specified by two text locations.
adsval	Array of udtAttributeDef	A list of attribute definitions.
asval	Array of udtAttribute	A list of attributes.
csval	Array of udtElementCatalogEntry	A list of element catalog entries.
isval	Array of F_Int	A list of integers.
uisval	Array of F_Int	A list of unsigned integers.
sval	F_String	A string value.
ival	F_Int	An integer value.
mval	F_Metric	A metric value.

udtAttribute

Details

The udtAttribute object represents a single attribute.

Properties

Property:	Type:	Description:
allow	F_Int	If True, allow error as special case
name	F_String	The attribute name
valflags	Array of F_Int	Validation error flags
values	Array of F_String	The list of attribute values

udtAttributeDef

Details

The udtAttributeDef object represents a single attribute definition.

The flags property determines whether the attribute is read-only, hidden, both, or neither. It can be one of the following values, or both values combined via a bitwise OR operation.

Value for flags	Meaning
mFV_AF_READ_ONLY	The attribute value is read-only
mFV_AF_HIDDEN	The attribute value is hidden
NULL	The attribute value is neither read-only nor hidden

The attrType property identifies the type for the attribute value. It can specify one of the following constatns:

attrType constant	Attribute value type
mFV_AT_STRING	Any arbitrary text string
mFV_AT_STRINGS	One or more arbitrary text strings
mFV_AT_CHOICES	A value from a list of choices
mFV_AT_INTEGER	A signed whole number (optionally restricted to a range of values)
mFV_AT_INTEGERS	One or more integers (optionally restricted to a range of values)
mFV_AT_REAL	A real number (optionally restricted to a range of values)
mFV_AT_REALS	One or more real numbers (optionally restricted to a range of values)
mFV_AT_UNIQUE_ID	A string that uniquely identifies the element
mFV_AT_UNIQUE_IDREF	A reference to a UniqueID attribute
mFV_AT_UNIQUE_IDREFS	One or more references to a UniqueID attribute

Properties

Property:	Type:	Description:
attrType	F_Int	The attribute type
choices	Array of F_String	The list of choices if the attribute type is mFV_AT_CHOICES
defValues	Array of F_String	Attribute default if the attribute is not required. If attribute is a type that can have multiple values, the default can have multiple strings.
flags	F_Int	Attribute is read-only, hidden, or neither. See the table below.
name	F_String	The attribute name
rangeMax	F_String	The maximum allowed value (if any).
rangeMin	F_String	The minimum allowed value (if any).
required	F_Int	True if the attribute is required.

udtCompareRet

Details

The udtCompareRet object is returned by the Compare method when comparing books or documents. It contains the IDs of the resulting summary and composite documents.

Properties

Property:	Type:	Description:
compId	F_Int	The ID of the composite document.
sumId	F_Int	The ID of the summary document.

udtElementCatalogEntry

Details

udtElementCatalogEntry objects represent entries in the Element Catalog. The flags property describes how an element of the specified type is to be validated. The flags value can be one of:

Flages constant	Meaning
mFV_STRICTLY_VALID	Catalog entry is strictly valid.

Flages constant	Meaning
mFV_LOOSELY_VALID	Catalog entry is loosely valid.
mFV_ALTERNATIVE	Catalog entry is an alternative.
mFV_INCLUSION	Catalog entry is valid because it is an inclusion.

Properties

Property:	Type:	Description:
flags	F_Int	The validation type.
objId	F_Int	The ID of the element definition.

udtElementLoc

Details

The udtElementLoc object specifies a location within a structural element. The parent element is the element that contains the location. The child element is the child of parent that immediately follows the location. The offset counts at what number of characters into parent to set the location.

For example, assume the following:

- A Para element that contains text and a List element
- The List element contains elements named Item
- The Item elements contain text

To set a location before any Item in List, parentId is the ID of the List element, childId is the ID of the given Item element in List, and offset is 0. To set the location after the last Item in List, parentId is the ID of the List element, childId is 0, and offset is 0. To set the location inside an Item, parentId is the ID of the Item element, childId is 0, and offset is the offset within the given Item element.

An udtElementRange object is two udtElementLoc objects named beg and end (see next). When the List element is selected, beg.parentId is Para, beg.childId is List,and the beg offset is 0; end.parentId is Para, end.childId is the next sibling of List (or 0 if List is the final element in Para), and end.offset is 0. If you select the root element, beg.parentId and end.parentId are both 0, beg.childId is the ID of the first child of the root and end.chidId is 0, and both offsets are 0.

Properties

Property:	Type:	Description:
childId	F_Int	Child element ID
offset	F_Int	Offset within parent element
parentId	F_Int	Parent element ID

udtElementRange

Details

The udtElementRange object represents a range of structured text. The range is expressed as beginning and end udtElementLoc objects.

Properties

Property:	Type:	Description:
beg	udtElementLoc	The location for the beginning of the element range
end	udtElementLoc	The location for the end of the element range

udtFont

Details

udtFont objects specify a combination of font characteristics. Each property specifies an index into a list of names in the current FMSession object. For example, family specifies the index of a name in the list of names specified by the session property FontFamilyNames. The weight property specifies the index of a name in the list of names specified by the session property FontWeightNames.

Properties

Property:	Type:	Description:
angle	F_Int	Index of the font angle
family	F_Int	Index of the font family
variation	F_Int	Index of the font variation
weight	F_Int	Index of the font weight

udtPoint

Details

The udtPoint object describes a pair of X, Y coordinates.

Properties

Property:	Type:	Description:
x	F_Metric	The X coordinate of the pair.
y	F_Metric	The Y coordinate of the pair.

udtPropIdent

Details

The udtPropIdent object provides a property identifier. Properties can be identified by either a name or a number (integer constant). FrameAC provides defined constants for property numbers (for example, mFP_Fill and mFP_Height). Only inset properties (facets) are identified by names.

If a property is identified by a name, the num is set to 0. If a property is identified by a number, udtPropIdentT.name is set to a null string.

Properties

Property:	Type:	Description:
name	F_String	The property name. If there is a value for number, this property is a NULL string.
num	F_Int	The property number. If there is a value for name, this property is set to zero.

udtPropVal

Details

The udtPropVal object describes a property/value pair.

Properties

Property:	Type:	Description:
propIdent	udtPropIdent	The property identifier.
propVal	udtTypedVal	The property value.

udtTab

Details

The udtTab object describes an individual tab. Note that the character specified by alignchar must be a single-byte character.

The type property can contain one of the following constants:

Type constant	Tab type
mFV_TAB_LEFT	Left tab
mFV_TAB_CENTER	Center tab
mFV_TAB_RIGHT	Right tab
mFV_TAB_DECIMAL	Decimal tab
mFV_TAB_RELATIVE_LEFT	Relative center tab (allowed only for format change lists)
mFV_TAB_RELATIVE_CENTER	Relative right tab (allowed only for format change lists)
mFV_TAB_RELATIVE_RIGHT	Relative decimal tab (allowed only for format change lists)
mFV_TAB_RELATIVE_DECIMAL	Relative center tab (allowed only for format change lists)

Properties

Property:	Type:	Description:
aligncharF_String		Character to align the tab around—for example, "." for decimal tabs.
leaderF_String		String of repeated characters to appear before the tab.
typeF_Int		Type of tab.
xF_Metric		Offset from the left margin.

udtTextItem

Details

The udtTextItem object represents an individual unit of text that is returned by the GetText or GetTextForRange method. These methods return an array of udtTextItem objects.

The udtTextItem object can represent:

- A string of characters with a common character format and condition tag
- The beginning or end of a line, paragraph, flow, column, or page
- An anchor for a table, footnote, marker, cross-reference, variable, or anchored frame
- A text properties change

The following table lists the values the udtTextItemT.dataType field can have and the types of data the corresponding text item provides.

Text item type (data type)	What the text item represents	Text item data
mFTI_TextObjId	The object that the offsets of all the text items are relative to	ID of an FMParagraph, FMCell, FMTextLine, FMTextInset_ApiClient, FMTextInset_Flow, FMTextInset_Text, or FMTextInset_TextTable
mFTI_String	A string of characters with the same condition and character format	F_String
mFTI_LineBegin	The beginning of a line	Nothing

Text item type (data type)	What the text item represents	Text item data
mFTI_LineEnd	The end of a line and the line-end type	If the line end is a normal line end, 0; if it is a forced line end, the mFTI_HardLineEnd flag is set; if it is a hyphen line end, the mFTI_HyphenLineEnd flag is set
mFTI_PgfBegin	The beginning of a paragraph	ID of an FMParagraph
mFTI_PgfEnd	The end of a paragraph	ID of an FMParagraph
mFTI_FlowBegin	The beginning of a flow	ID of an FMFlow
mFTI_FlowEnd	The end of a flow	ID of an FMFlow
mFTI_PageBegin	The beginning of a page	ID of an FMPage
mFTI_PageEnd	The end of a page	ID of an FMPage
mFTI_SubColBegin	The beginning of a column	ID of an FMSubCol
mFTI_SubColEnd	The end of a column	ID of an FMSubCol
mFTI_FrameAnchor	An anchored frame	ID of an FMAnchoredFrame
mFTI_FnAnchor	A footnote	ID of an FMFootnote
mFTI_TblAnchor	A table	ID of an FMTable
mFTI_MarkerAnchor	A marker	ID of an FMMarker
mFTI_XRefBegin	The beginning of a cross-reference	ID of an FMCrossReference
mFTI_XRefEnd	The end of a cross-reference	ID of an FMCrossReference
mFTI_TextFrameBegin	The beginning of a text frame	ID of an FMTextFrame
mFTI_TextFrameEnd	The end of a text frame	ID of an FMTextFrame
mFTI_VarBegin	The beginning of a variable	ID of an FMVariable
mFTI_VarEnd	The end of a variable	ID of an FMVariable
mFTI_ElementBegin	The beginning of a structural container element	ID of an FMElement
mFTI_ElementEnd	The end of a structural container element	ID of an FMElement
mFTI_ElemPrefixBegin	The beginning of an element's prefix	ID of an FMElement
mFTI_ElemPrefixEnd	The end of an element's prefix	ID of an FMElement
mFTI_ElemSuffixBegin	The beginning of an element's suffix	ID of an FMElement
mFTI_ElemSuffixEnd	The end of an element's suffix	ID of an FMElement

Text item type (data type)	What the text item represents	Text item data
mFTI_CharProps Change	A change in the text properties	Flags indicating which properties have changed (see the table below)
mFTI_RubiComposite Begin	The beginning of a rubi composite (and the beginning of oyamoji text).	ID of an FMRubi
mFTI_RubiComposite End	The end of a rubi composite.	ID of an FMRubi
mFTI_RubiTextBegin	The beginning of rubi text (and the end of oyamoji text).	ID of an FMRubi
mFTI_RubiTextEnd	The end of rubi text.	ID of an FMRubi

The following table lists the bit flags that your code can combine with the idata field of an mFTI_CharPropsChange text item via bitwise AND operations. For example, to determine if the font family changed, bitwise AND the mFTF_FAMILY flag with the idata field.

Flag	Meaning
mFTF_FAMILY	The font family has changed.
mFTF_VARIATION	The font variation has changed.
mFTF_WEIGHT	The font weight has changed.
mFTF_ANGLE	The font angle has changed.
mFTF_UNDERLINING	The underlining has changed.
mFTF_STRIKETHROUGH	The strikethrough characteristic has changed.
mFTF_OVERLINE	The overline characteristic has changed.
mFTF_CHANGEBAR	The change bars have changed.
mFTF_OUTLINE	The outline characteristic has changed.
mFTF_SHADOW	The shadow characteristic has changed.
mFTF_PAIRKERN	The pair kerning has changed.
mFTF_SIZE	The font size has changed.
mFTF_KERNX	The kern-x characteristic has changed.
mFTF_KERNY	The kern-y characteristic has changed.
mFTF_SPREAD	The font spread has changed.
mFTF_COLOR	The color has changed.
mFTF_CHARTAG	The Character Catalog format has changed.
mFTF_CAPITALIZATION	The capitalization has changed.

Flag	Meaning
mFTF_POSITION	The character position has changed.
mFTF_CONDITIONTAG	The condition tag has changed.
mFTF_STRETCH	Font stretch value has changed
mFTF_LANGUAGE	Character language has changed
mFTF_TSUME	Tsume setting has changed
mFTF_IIF	Inline input has changed
mFTF_ENCODING	The text encoding has changed.
mFTF_ALL	OR of all the flags listed above.

Properties

Property:	Type:	Description:
dataType	F_Int	The type text item, represented by an mFTI_ constant.
idata	F_Int	If the text item is for a FrameMaker document object, the ID of that object.
offset	F_Int	Offset from the paragraph or text line beginning.
sdata	F_String	If type is mFTI_String, the corresponding text string.

udtTextLoc

Details

The udtTextLoc object specifies a location within the text of an FMParagraph or an FMTextLine.

Properties

Property:	Type:	Description:
objId	F_Int	The FMParagraph or an FMTextLine.
offset	F_Int	The offset of character from the beginning of the object.

udtTextRange

Details

The udtTextRange object specifies a text range. A text range can span paragraphs. It can't span graphic text lines or flows. A text range is expressed as two udtTextLoc objects—a beg udtTextLoc and an end udtTextLoc.

Note that the offset properties for the beg and end udtTextLoc objects can specify offsets relative to the beginning of a paragraph or text line. These offsets can also use the special value mFV_OBJ_END_OFFSET. mFV_OBJ_END_OFFSET specifies the offset of the last character in the object containing the text range.

Properties

Property:	Type:	Description:
beg	udtTextLoc	The location of the beginning of the text range.
end	udtTextLoc	The location of the end of the text range.

udtTypedVal

Details

The udtTypedVal object is a general object that can represent data of many different types.

The valType property indicates the type of value the object provides—that is, which property of the F_TypedValValues object to use. The constants used in the valType property are described in the following table:

valType constant	Value data type	mF_TypedValValues property
mFT_AttributeDefs	Array of udtAttributeDef	adsval
mFT_Attributes	Array of udtAttribute	asval
mFT_Integer	F_Int	ival

valType constant	Value data type	mF_TypedValValues property
mFT_Ints	Array of F_Int	isval
mFT_Metric	F_Metric	mval
mFT_Metrics	Array of F_Metric	msval
mFT_String	F_String	sval
mFT_Strings	Array of F_String	ssval
mFT_Id	F_Int	ival
mFT_Points	Array of udtPoint	psval
mFT_Tabs	Array of udtTab	tsval
mFT_TextLoc	udtTextLoc	tlval
mFT_TextRange	udtTextRange	trval
mFT_UInts	Array of F_Int	uisval
mFT_ElementCatalog	Array of udtElementCatalogEntry	csval
mFT_UBytes	Array of F_Int	

Properties

Property:	Type:	Description:
value	F_TypedValValues	A value of any of the types represented by the F_TypedValValues object.
valType	F_Int	A constant indicating the value type.

A Special Cases for Object Properties

A number of object properties have interdependencies, large lists of possible values, or other qualities that require extra explanation. This appendix provides the additional information you may need to use these properties effectively.

FMDocument Hypertext Properties

When you use the HypertextCommandText and HypertextDoValidate properties to parse and validate a hypertext command, FrameMaker sets specific values to the other document hypertext properties. A number of these values can be defined integer values. The following tables list the integer values and their meanings.

HypertextParseError

The following table shows error codes to which FP_HypertextParseErr can be set:

Value	Meaning
mFV_HypertextSyntaxOK	No parse errors
mFV_HypertextEmptyCommand	Hypertext string is empty
mFV_Hypertext UnrecognizedCommand	Cannot map the first keyword to an existing FP_HypertextParsedCmdCode
mFV_HypertextMissingArguments	One or more arguments required for the command is missing
mFV_HypertextExtraArguments	More than the required number of arguments for the command; extra arguments were ignored
mFV_HypertextBadSyntax PathSpec	File reference expected for this command, but no valid filepath found
mFV_HypertextUnanchoredPartialPath	File reference is relative to the current document, but the current document has not been saved; file location could not be calculated
mFV_HypertextHelpDirNotFound	Default help directory either does not exist (help was not installed) or cannot be found
mFV_HypertextExpectedANumberParam	Command expected a number but got text; check FP_HypertextParseBadParam

HypertextValidateError

The following table shows error codes to which HypertextValidateErr can be set:

Value	Meaning
mFV_HypertextValid	No validation errors
mFV_HypertextUsesDefaultText	Default text was found as an argument; are you sure the default text is what you want?
mFV_HypertextFileNotRegular	The referenced file could not be found, or is not a regular file; for example, it could be a directory name
mFV_HypertextFileNotMakerDoc	The referenced file is not made by a FrameMaker product
mFV_HypertextCantOpenDestFile	Can't open the file; perhaps you don't have permission, or the file is locked
mFV_HypertextDestinationLinkNotFound	The referenced file is valid, but can't find the named link within it
mFV_HypertextPageNameNotFound	The referenced file is valid, but can't find the specified page
mFV_HypertextUnrecognizedObjectType	The referenced file is valid, but the link is to an object with an unrecognized object type
mFV_HypertextObjectIDNotFound	A link to an object, but can't find the object
mFV_HypertextBadMatrixSize	One or both of the matrix dimensions is bad; must be between 1 and 99
mFV_HypertextMatrixCommandInvalid	One of the cammands in the reference page flow for a matrix command has a parse or validation error
mFV_HypertextFlowMissingLines	The reference flow for a matrix or popup command is missing one or more lines
mFV_HypertextNoNamedFlow	Can't find the named reference flow for a matrix or popup command
mFV_HypertextRecursiveFlow	The reference flow for a matrix or popup command contains nested popup or matrix commands that name a parent reference flow
mFV_HypertextMissingPopupMarker	At least one entry in the popup command's reference flow has no hypertext marker in it
mFV_HypertextMissingPopupLabelItem	One entry in the popup command's reference flow has no text in it

Value	Meaning
mFV_HypertextEmptyLineIn MiddleOfPopup	One entry in the popup command's reference flow has no text in it
mFV_HypertextCommand IllegalWithinPopup	Invalid command in the popup command's reference flow; for example, matrix or newlink
mFV_HypertextFcodeInvalid	Invalid FCode in the hypertext command

Command Codes

The following table shows error codes to which HypertextParsedCmdCode can be set:

Value	Meaning
mFV_CmdError	Parser is in an error state
mFV_CmdUnknown	Unknown command
mFV_CmdNoop	Command causes no event
mFV_CmdAlert	alert command
mFV_CmdAlertTitle	alerttitle command
mFV_CmdExit	exit commant
mFV_CmdGoToLink	gotolink command
mFV_CmdGoToLinkFitWin	gotolinkfitwin command
mFV_CmdGoToNew	gotonew command
mFV_CmdGoToPage	gotopage command
mFV_CmdGoToObjectId	gotoObjectId command
mFV_CmdGoToObjectIdFitWin	gotoObjectIdfitwin command
mFV_CmdMatrix	matrix command
mFV_CmdMessage	message command
mFV_CmdNewLink	newlink command
mFV_CmdNextPage	nextpage command
mFV_CmdOpenLink	openlink command
mFV_CmdOpenLinkFitWin	openlinkfitwin command
mFV_CmdOpenNew	opennew command
mFV_CmdOpenObjectId	openObjectId command
mFV_CmdOpenObjectIdFitWin	openObjectIdfitwin command
mFV_CmdOpenPage	openpage command
mFV_CmdPopup	popup command
mFV_CmdPreviousLink	previouslink command

Value	Meaning
mFV_CmdPreviousLinkFitWin	previouslinkfitwin command
mFV_CmdPreviousPage	previouspage command
mFV_CmdQuit	quit command
mFV_CmdQuitAll	quitall command

Link Destinations

The following table shows error codes to which HypertextParsedCmdDest can be set:

Value	Meaning
mFV_DestNowhere	No destination found
mFV_DestMarkerNewLink	Destination is a newlink
mFV_DestFirstPage	Destination is the first page of a file
mFV_DestLastPage	Destination is the last page of a file
mFV_DestPageNum	Destination is a named page (usually a page number)
mFV_DestFluidFlow	Destination is to a fluid flow document
mFV_DestMarker	Destination is a marker
mFV_DestObjectId	Destination is an object ID (usually for generated hypertext commands)
mFV_DestXRef	Destination is a cross-reference

Link Destination Object Types

The following table shows error codes to which HypertextParsedCmdDestObjType can be set:

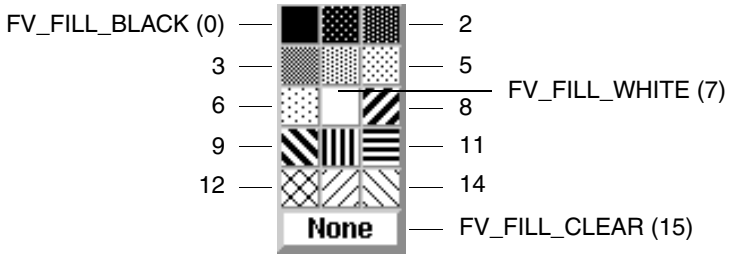
Value	Meaning
mFV_ObjectUnknown	Unknown or invalid object
mFV_ObjectMarker	Object is a marker
mFV_ObjectPgf	Object is a paragraph
mFV_ObjectXref	Object is a cross-reference
mFV_ObjectGraphic	Object is a graphic
mFV_ObjectElement	Object is an element
mFV_ObjectTextInset	Object is a text inset
mFV_ObjectDataLink	Object is subscribed data

FMGraphic properties

This section provides information about various FMGraphic properties and how to use them.

Pen and FIll properties

Pen and Fill patterns are represented by an integer. A Pen pattern can be a number from 0 to 7, and a Fill pattern can be a number from 0 to 15. The figure below shows how these values correspond to the patterns you can select in the FrameMaker Graphics Toolbox.



Dash Patterns

The Dash property in the FMGraphic object specifies a dash pattern that is repeated for the length of an object's border. The pattern is stored in an F_MetricsT structure. The 0th element of the F_MetricsT.F_MetricsT_val array stores the length of the first dash; the 1st element stores the following space; the 2nd element stores the next dash; and so on for an even number of elements.

Graphic and Text Inset Hint Strings

FrameMaker documents can included graphics and text that are imported into the document by reference. These imported files are stored in the document as insets—FMInset for an imported graphic file, and FMTextInset for imported text. When you import one of these files via FrameAC, you must specify an import hint string. This is a string that identifies the data format so FrameMaker can invoke the necessary filter and convert that data format into a usable form.

Hint strings for graphic insets and text insets are very similar. However, each type of hint string has particular issues. Both types of hint strings are discussed below.

Graphic Inset Import Hint Strings

The FP_ImportHint property of an FMInset object specifies a record that identifies the filter used to import a graphic. FrameMaker uses this record to find the correct filter to reimport the graphic when the document is reopened or the inset is manually updated.

The ImportHint property does not apply to graphics imported by copy. FrameMaker uses the facet name stored with the graphic to identify the filter that filtered a graphic imported by copy.

The syntax of the record specified by the ImportHint property is:

```
record_vers vendor format_id platform filter_vers filter_name
```

Note that the fields in the record are not separated by spaces—spaces are significant data except those that appear in the filter_name field. For example:

```
0001PGRFPICTMAC61.0 Built-in PICT reader
0001ORBLSVGIAll Platform SVG reader for 7.0
```

Each field of the record (except filter_name) specifies a four-byte code. If a code contains fewer than four alphanumeric characters, the remaining bytes must be filled out with spaces.

The rest of this section describes each field in the record.

`record_vers` specifies the version of the record, currently `0001`.

`vendor` is a code specifying the filter's vendor. The code is a string of four characters. The following table lists the possible codes

Code	Meaning
`PGRF`	Built-in Frame filters
`FAPI`	External Frame FDK client filter
`FFLT`	External Frame filters
`IMAG`	External ImageMark filters
`XTND`	External XTND filters

This is not a comprehensive list of codes. Codes may be added to this list by Frame or by developers at your site.

`format_id` is a code specifying the format that the filter translates. The code is a string of four characters. The following table lists some of the possible codes.

Code	Meaning
`CDR`	CorelDRAW
`CGM`	Computer Graphics Metafile
`DIB`	Device-independent bitmap (Windows)
`DRW`	Micrografx CAD
`DXF`	Autodesk Drawing eXchange file (CAD files)
`EMF`	Enhanced Metafile (Windows)
`EPSB`	Encapsulated PostScript Binary (Windows)
`EPSD`	Encapsulated PostScript with Desktop Control Separations (DCS)

Code	Meaning
EPSF	Encapsulated PostScript (Macintosh)
EPSI	Encapsulated PostScript Interchange
FRMI	FrameImage
FRMV	FrameVector
G4IM	CCITT Group 4 to Image
GEM	GEM file (Windows)
GIF	Graphics Interchange Format (Compuserve)
HPGL	Hewlett-Packard Graphics Language
IGES	Initial Graphics Exchange Specification (CAD files)
IMG4	Image to CCITT Group 4 (UNIX)
MooV	QuickTime Movie
OLE	Object Linking and Embedding Client (Microsoft)
PCX	PC Paintbrush
PICT	QuickDraw PICT
PNTG	MacPaint
SNRF	Sun Raster File
SRGB	SGI RGB
SVGI	Scalable Vector Graphics
TIFF	Tag Image File Format
WMF	Windows Metafile
WPG	WordPerfect Graphics
XWD	X Windows System Window Dump file

`platform` is a code specifying the platform on which the filter was run. The code is a string of four characters. The following table lists the possible codes.

Code	Meaning
MAC6	Macintosh 68000 series
MACP	Power Macintosh‘
WINT	Windows NT
WIN3	Windows 3.1
WIN4	Windows 95
OS/2	IBM OS/2
UNIX	Generic X/11 (Sun, HP)

filter_vers is a string of four characters identifying the version of the filter on that platform. For example, version 1.0 of a filter is represented by the string 1.0.

filter_name is a text string (up to 31 characters long) that describes the filter.

Text Inset Import Hint Strings

The ImportHint property specifies a record that identifies the filter used to import text by reference. FrameMaker uses this record to find the correct filter to update a text inset. The syntax of this record is:

```
record_vers vendor format_id platform filter_vers filter_name
```

Note that the fields in the record are not separated by spaces.

```
0001XTNDWDBNMACP0002MS Word 4,5
```

You can determine which ImportHint strings are registered by querying the ImportFilters property of the FMSession object.

Each field of the record (except filter_name) specifies a four-byte code. If a code contains fewer than four alphanumeric characters, the remaining bytes must be filled out with spaces.

The rest of this section describes each field in the record.

record_vers specifies the version of the record, currently 0001.

vendor is a code specifying the filter's vendor. The code is a string of four characters. The following table lists some possible codes.

Code	Meaning
PGRF	Built-in Frame filters
FAPI	External Frame FDK client filter
FFLT	External Frame filters
IMAG	External ImageMark filters
XTND	External XTND filters

format_id is a code specifying the format that the filter translates. The code is a string of four characters. The following table lists the possible codes.

Code	Meaning
HTML	
XML	
WDBN	
WPBN	HTML document (for export, only)
RTF	XML document (for export, only)
IAF	Microsoft Word compound document

Code	Meaning
MRTF	WordPerfect compound document
MIAF	Microsoft's RTF compound document
MWPB	Interleaf compound document
MML	MIF to RTF export
CVBN	MIF to IAF export
DCA	MIF to WordPerfect export
TRFF	Maker Markup Language
TRFA	Corel Ventura compound document (Windows)
TRFE	DCA to MIF (UNIX)
TRFS	troff to MIF (UNIX only)
CVBN	troff -man to MIF (UNIX only)
TEXT	troff -me to MIF (UNIX only)
TXIS	troff -ms to MIF (UNIX only)
TANS	Plain text
TMAC	Text ISO Latin 1
TASC	Text Roman 8 (Hewlett Packard unix)
TJIS	Text ANSI (Windows)
TSJS	Text (Macintosh)
TEUJ	Text ASCII
TBG5	Japanese JIS
TEUH	Japanese Shift-JIS
TXHZ	Japanese EUC
TXGB	Traditional Chinese BIG 5
TKOR	Traditional Chinese EUC-CNS

Note that this is not a comprehensive list of codes. Codes may be added to this list by Frame or by developers at your site

`platform` is a code specifying the platform on which the filter was run. The code is a string of four characters. The following table lists the possible codes.

Code	Meaning
MAC6	Macintosh 68000 series
MACP	Power Macintosh‘
WINT	Windows NT
WIN3	Windows 3.1

Code	Meaning
WIN4	Windows 95
OS/2	IBM OS/2
UNIX	Generic X/11 (Sun, HP)

filter_vers is a string of four characters identifying the version of the filter on that platform. For example, version 1.0 of a filter is represented by the string 1.0.

filter_name is a text string (up to 31 characters long) that describes the filter.

For example, the following two records specify the HTML and XML filters on the Windows NT platform:

```
"0001ADBEHTMLWINT0001HTML"

"0001ADBEXMLWINT0001XML "
```

B FrameMaker Document and Session Architecture

Some FrameMaker data and constructs require additional explanations to describe how they fit into the overall structure of a FrameMaker session or document. This appendix discusses these issues so you can work with these constructs effectively.

How FrameMaker Represents Text

FrameAC represents the text in each paragraph or graphic text line with a list of udtTextItem data objects. Each udtTextItem consists of:

- offset — the number of characters from the beginning of the queried object
- dataType — the type of text item, for example, mFTI_String
- value — can be the ID of an object or a string of text, depending on the dataType

:The offset value specifies the distance between the start of the text item and the beginning of the text line or paragraph. This distance is measured in the number of characters (both regular characters and anchor symbols).

Each of the following constitutes a separate text item:

- A string of characters with common text properties

  A text item can contain a string that is as long as a line of text. However, the API uses a separate text item for each section of the text that has different text properties. If a single property (such as the font weight, font angle, or condition format) is different, the API starts a new text item. So a single line of text may require several text items to represent it.

- The beginning or end of a line, paragraph, flow, column, page, or structural element

  The API uses text items to indicate the beginning or end of the various entities that organize text. Most of these text items specify the ID of an object. Text items that indicate the end of a line specify whether the line end is a regular, hyphenated, or hard line end.

- An anchor for a table, footnote, marker, cross-reference, variable, or anchored frame

  The API represents tables, footnotes, markers, cross-references, variables, and anchored frames with separate objects. It uses a text item to represent the anchor for each of these entities. The text item specifies the ID of the object. For example, the API represents a table with an FO_Tbl object. It uses a table anchor (FTI_TblAnchor) text item to indicate where the table occurs in the text.

- A text properties change

This type of text item identifies the point in text at which the text properties change. It specifies flags that indicate which text properties differ from the properties of the text immediately preceding the text item.

The following table lists the values the `F_TextItemT.dataType` field can have and the types of data the corresponding text item provides.

Text item type (dataType)	What the text item represents	Text item data
`mFTI_CharPropsChange`	A change in the text properties	Flags indicating which properties have changed (see the table below)
`mFTI_ElementBegin`	The beginning of a container element	ID of an FMElement
`mFTI_ElementEnd`	The end of a container element	ID of an FMElement
`mFTI_ElemPrefixBegin`	The beginning of an element's prefix	ID of an FMElement
`mFTI_ElemPrefixEnd`	The end of an element's prefix	ID of an FMElement
`mFTI_ElemSuffixBegin`	The beginning of an element's suffix	ID of an FMElement
`mFTI_ElemSuffixEnd`	The end of an element's suffix	ID of an FMElement
`mFTI_FlowBegin`	The beginning of a flow	ID of an FMFlow
`mFTI_FlowEnd`	The end of a flow	ID of an FMFlow
`mFTI_FnAnchor`	A footnote	ID of an FMFootnote
`mFTI_FrameAnchor`	An anchored frame	ID of an FMAnchoredFrame
`mFTI_LineBegin`	The beginning of a line	Nothing
`mFTI_LineEnd`	The end of a line and the line end type	If the line end is a normal line end, `0`; if it is a forced line end, the `mFTI_HardLineEnd` flag is set; if it is a hyphen line end, the `mFTI_HyphenLineEnd` flag is set
`mFTI_MarkerAnchor`	A marker	ID of an FMMarker
`mFTI_PageBegin`	The beginning of a page	ID of an FMBodyPage, FMHiddenPage, FMMasterPage, FMReferencePage
`mFTI_PageEnd`	The end of a page	of an FMBodyPage, FMHiddenPage, FMMasterPage, FMReferencePage

Text item type (dataType)	What the text item represents	Text item data
mFTI_PgfBegin	The beginning of a paragraph	ID of an FMParagraph
mFTI_PgfEnd	The end of a paragraph	ID of an FMParagraph
mFTI_RubiCompositeBegin	The beginning of a rubi composite (and the beginning of oyamoji text).	ID of an FMRubi
mFTI_RubiCompositeEnd	The end of a rubi composite.	ID of an FMRubi
mFTI_RubiTextBegin	The beginning of rubi text (and the end of oyamoji text).	ID of an FMRubi
mFTI_RubiTextEnd	The end of rubi text.	ID of an FMRubi
mFTI_String	A string of characters with the same condition and character format	A character string
mFTI_SubColBegin	The beginning of a column	ID of an FMSubCol
mFTI_SubColEnd	The end of a column	ID of an FMSubCol
mFTI_TblAnchor	A table	ID of an FMTable
mFTI_TextFrameBegin	The beginning of a text frame	ID of an FO_TextFrame
mFTI_TextFrameEnd	The end of a text frame	ID of an FMTextFrame
mFTI_TextInsetBegin	The beginning of a text inset	ID of an FMTextInset_ApiClient, FMTextInset_Flow, FMTextInset_Text, or FMTextInset_TextTable
mFTI_TextInsetEnd	The end of a text inset	ID of an FMTextInset_ApiClient, FMTextInset_Flow, FMTextInset_Text, or FMTextInset_TextTable
mFTI_TextObjId	The object to which the offsets of all the text items are relative	ID of an FMParagraph, FMCell, FMTextLine, FMTextInset_ApiClient, FMTextInset_Flow, FMTextInset_Text, or FMTextInset_TextTable
mFTI_VarBegin	The beginning of a variable instance	ID of an FMVariable
mFTI_VarEnd	The end of a variable instance	ID of an FMVariable
mFTI_XRefBegin	The beginning of a cross-reference instance	ID of an FMCrossReference

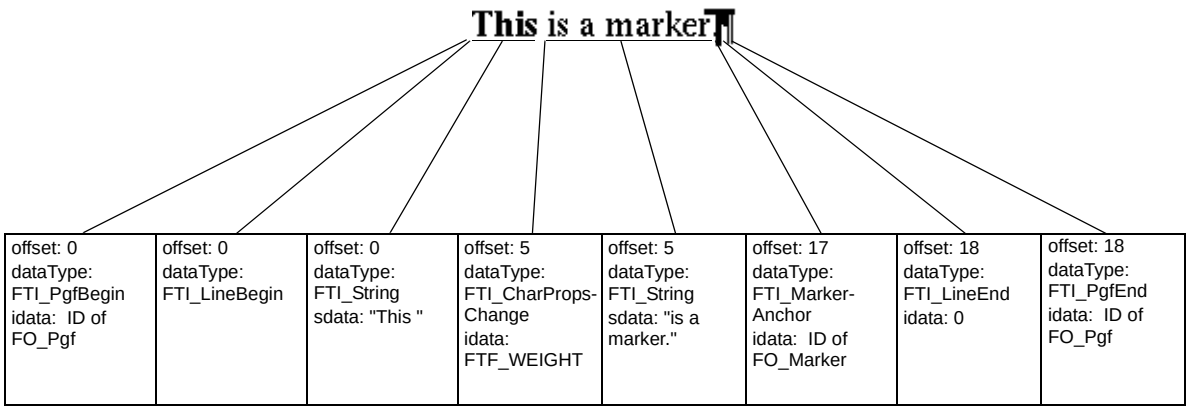
Text item type (dataType)	What the text item represents	Text item data
mFTI_XRefEnd	The end of a cross-reference instance	ID of an FMCrossReference

The following table lists the bit flags that a client can bitwise AND with the text item data for an mFTI_CharPropsChange text item. The table also shows the types of text property changes each flag indicates. For example, to determine if the font family changed, bitwise AND the mFTF_FAMILY flag with the text item data.

Flag	Meaning
mFTF_ALL	OR of all the flags listed above.
mFTF_ANGLE	The font angle has changed.
mFTF_CAPITALIZATION	The capitalization has changed.
mFTF_CHANGEBAR	The change bars have changed.
mFTF_CHARTAG	The Character Catalog format has changed.
mFTF_COLOR	The color has changed.
mFTF_CONDITIONTAG	The condition tag has changed.
mFTF_ENCODING	The text encoding has changed.
mFTF_FAMILY	The font family has changed.
mFTF_IIF	An internal flag having to do with asian text. input. If there is a non-zero value for this flag, a front end processor is controlling that text; you should not modify the associated text item.
mFTF_KERNX	The kern-x characteristic has changed.
mFTF_KERNY	The kern-y characteristic has changed.
mFTF_LANGUAGE	Character language has changed
mFTF_OUTLINE	The outline characteristic has changed.
mFTF_OVERLINE	The overline characteristic has changed.
mFTF_PAIRKERN	The pair kerning has changed.
mFTF_POSITION	The character position has changed.
mFTF_SHADOW	The shadow characteristic has changed.
mFTF_SIZE	The font size has changed.
mFTF_SPREAD	The font spread has changed.
mFTF_STRETCH	Font stretch value has changed
mFTF_STRIKETHROUGH	The strikethrough characteristic has changed.
mFTF_TSUME	Tsume setting has changed
mFTF_UNDERLINING	The underlining has changed.

Flag	Meaning
mFTF_VARIATION	The font variation has changed.
mFTF_WEIGHT	The font weight has changed.

The following figure represents a paragraph and the text items that represent the paragraph's text.


This is a marker.
offset: 0 dataType: FTI_PgfBegin idata: ID of FO_Pgf
offset: 0 dataType: FTI_LineBegin
offset: 0 dataType: FTI_String sdata: "This "
offset: 5 dataType: FTI_CharProps-Change idata: FTF_WEIGHT
offset: 5 dataType: FTI_String sdata: "is a marker."
offset: 17 dataType: FTI_Marker-Anchor idata: ID of FO_Marker
offset: 18 dataType: FTI_LineEnd idata: 0
offset: 18 dataType: FTI_PgfEnd idata: ID of FO_Pgf

There are several important things to note about the text items in this figure:

- Because the string `"This "` and the string `"is a marker."` have different font weights, there are separate text items for them.
- The `mFTI_CharPropsChange` text item indicates that the text properties have changed; the `mFTF_WEIGHT` flag that it specifies indicates that the font weight has changed.
- The marker anchor is counted in the offset.

How FrameAC Represents Font Information

FO_Session objects have properties (such as FP_FontFamilyNames) that provide arrays of the names of the font families, variations, angles, and weights available in the current session. These lists are referenced by F_StringsT objects.

For example, if Bold and Regular are the only font weights available in the current session, the fields of the F_StringsT structure specified by the FO_Session property, FP_FontWeightNames, have the following values:

len: 3
val: {"<None>","Regular","Bold"}

To set a character format's weight to Bold in this session, you set its FP_FontWeight property to 2.

You can also use the following properties to specify a font:

- FP_FontPlatformName specifies a font name that uniquely identifies the font on a specific platform.
- FP_FontPostScriptName specifies the name given to a font when it is sent to a PostScript printer (specifically, the name that is passed to the PostScript FindFont operator before any font coordination operations).

  The PostScript name is unique for all PostScript fonts, but may not be available for fonts that have no PostScript version.

  For the same Asian font, the PostScript name can be different on different platforms. This is because they might have slightly different extensions to the character mappings. For example, Ryumin-Light on the Macintosh is Ryumin-Light-83pv-RKSJ-H, while it is Ryumin-Light-90ms-RKSJ-H on a Windows system. Since these are the same fonts, FrameMaker will treat these as the same PostScript name. To do so, FrameMaker ignores the following keywords in PostScript names:

  -83pv

  -90pv

  -90ms

  -Ext

  -Add

  -NWP

The FP_FontPlatformName property specifies a platform-specific ASCII string that uniquely identifies a font for a particular platform. The string consists of several fields separated by periods.

The following strings are valid representations of the Windows font, Helvetica Narrow Bold Oblique:

W.Helvetica-Narrow.I.700
W.Helvetica.I.700.Narrow

When reading in a document, FrameMaker+SGML determines a font name by checking font properties in the following order:

- FP_FontPlatformName
- Combination of FP_FontFamily, FP_FontVariation, FP_FontWeight, and FP_FontAngle
- FP_FontPostScriptName

Your FrameAC programs do not need to use all three methods to change fonts. You should always specify the PostScript name, if it is available. You can safely use the PlatformName because FrameAC programs always run on the Windows platform.

Standard Marker Types for a Document

Every document includes a set of required marker types; Header/Footer $1, Header/Footer $2, Index, Comment, Subject, Author, Glossary, Equation, Hypertext Cross-Ref, and Conditional Text. These are required markers, and cannot be deleted.

You can add an existing public marker type to the standard list by setting the name string to the AddMarkerTypeToStandardMarkers property of the current session object. Once you add a marker type to this list, it remains for the entire session; you must quit the session to remove it.

For example, you can set the OldTypeNum property to `17`, and set the AddMarkerTypeToStandardMarkers property to `"MyMarkerType"`. This first sets a session integer for OldTypeNum to `17` so that for the rest of the current session, markers of type 17 (from earlier documents) will come into new documents as markers of type `MyMarkerType`.

If the OldTypeNum you specify is taken, your new marker type will not be added to the list of standard marker types. To confirm that your marker type was added to the standard list, get the MarkerNames property from the SessionId object.

It's possible for the OldTypeNum you specified to be taken; another API client may have already used that value when assigning a marker type to the standard list. For example, HTML export in FrameMaker 5.5 is performed by a client that adds the HTML Macro marker type to the standard list. The value of that marker's OldTypeNum is 11. After that client is initialized, no other clients can use the same value for OldTypeNum when assigning a marker to the standard list.

XMP Data in FrameMaker Documents

FrameMaker supports XMP Metadata, which is a protocol to store information about a file as encoded packets that are available to external applications. This information is similar to the information stored in the PDF Document Info dictionary. However, XMP data can contain fields that have no counterpart in PDF Document Info.

FrameMaker maps the values of string pairs in the PDFDocInfo property to XMP Metadata. If an external application modifies the document's metadata, these mapped fields will be updated in PDFDocInfo. Likewise, if you change a field in PDFDocInfo via the FDK, then FrameMaker+SGML will update the encoded XMP packets to reflect this new information. The following table lists the supported fields. External indicates that the field value can be

changed by your client, an external application, or the user interface. Internal indicates a field that FrameMaker+SGML maintains—you cannot modify its value

PDF Field	XMP Field	Internal/External
Author	dc:Creator	External
Title	dc:Title	External
Subject	dc:Description	External
Keywords	pdf:Keywords	External
Copyright	dc:Rights	External
Web Statement	xapRights:WebStatement	External
Job Reference	xapBJ:JobRef	External
Marked	xapRights:Marked	External
Creation Date	xap:CreateDate	Internal
ModDate	xap:ModifyDate	Internal
Metadata Date	xap:MetadataDate	Internal
Creator	pdf:CreatorTool	Internal (FrameMaker—not displayed in dialog box)

You can modify XMP data directly for a document by setting a value to the FileInfoPacket document property. The FDK sample clients include a client that reads a text file and sets the file's content to the FilePacket property. XMP uses the RDF syntax—see http://www.w3.org/1999/02/22-rdf-syntax-ns#, or print the FilePacket to the console to see the XMP syntax. XMP data can include UNICODE characters.

The PDFDocInfo property defines a list of strings to enter in a PDF Document Info dictionary. For one dictionary entry you provide two strings; the first is the entry name, and the second is the entry content. The entry name can be up to 126 bytes; the entry content can be up to 32765 characters.

The entry name is a string of bytes within the ASCII range. For non-printable ASCII, provide Hex codes. To represent a Hex code, precede the code with the "#" character; for example #24. To represent the "#" character, enter #23. Finally, an entry name may not include a byte with a value of zero (#00).

The entry content can include Unicode encoding. However, FrameMaker doesn't support support Unicode in these strings. To enter Unicode characters in these strings, you must use an ASCII representation of Unicode that the FrameMaker product understands, as the following table shows:

To represent these characters:	Use:
Unicode characters within the standard ASCII range, with the exception of the ampersand (&) character	Standard ASCII characters

To represent these characters:	Use:
Unicode characters outside the standard ASCII range	A token (&#x) to identify a Unicode character followed by the hexadecimal value of the Unicode character; for example, &#xC2A7
Ampersand	&#x0026

PDF Document Info Dictionaries

The FP_PDFDocInfo property defines a list of strings to enter in a PDF Document Info dictionary. For one dictionary entry you provide two strings; the first is the entry name, and the second is the entry content. The entry name can be up to 126 bytes; the entry content can be up to 32765 characters.

The entry name is a string of bytes within the ASCII range. For non-printable ASCII, provide Hex codes. To represent a Hex code, precede the code with the “#” character; for example #24. To represent the “#” character, enter #23. Finally, an entry name may not include a byte with a value of zero (#00).

The entry content can include Unicode encoding. However, FrameMaker doesn’t support support Unicode in these strings. To enter Unicode characters in these strings, you must use an ASCII representation of Unicode that the FrameMaker product understands, as the following table shows:

To represent these characters:	Use:
Unicode characters within the standard ASCII range, with the exception of the ampersand (&) character	Standard ASCII characters
Unicode characters outside the standard ASCII range	A token (&#x) to identify a Unicode character followed by the hexadecimal value of the Unicode character; for example, &#xC2A7
Ampersand	&#x0026

For more information on Acrobat specific functionality see the Acrobat Documentation.